

فهرست مطالب

چکیده ----- ۵

تاریخچه ----- ۷

فصل اول: سیستم عامل و انواع آن

۱-۱ مقدمه ای بر ایجاد سیستم عامل ها ----- ۱۰

۱-۱-۱ سیستم عامل چه کاری انجام می دهد. ----- ۱۰

۱-۲ انواع سیستم عامل: ----- ۱۰

۱-۳ وظایف سیستم عامل چیست؟ ----- ۱۱

۱-۴ گوشی های بدون سیستم عامل ، گوشی های جاوا: ----- ۱۴

۱-۵ انواع سیستم عامل های موبایل: ----- ۱۵

فصل دوم: آشنایی با سیستم عامل IOS

۲-۱ IOS ----- ۱۷

۲-۲ ویژگی ها ----- ۱۸

۲-۲-۱ صفحه اصلی ----- ۱۸

۲-۳ مرکز هشدار دهنده (Notification Center) ----- ۱۸

۲-۴ پوشه ها (Folders) ----- ۱۸

۲-۵ عملکرد چند تکلیفی (Multitasking) ----- ۱۹

۲-۶ راه گزینی برنامه های کاربردی (Switching applications) ----- ۱۹

۲-۷ مرکز بازی (Game Center) ----- ۲۰

۲-۸ محیط برنامه نویسی ----- ۲۰

۲۲	-----	UNLOCKING ۲-۹
۲۲	-----	Digital right management ۲-۱۰
۲۴	-----	۲-۱۱ معماری سیستم عامل Phone :
۲۵	-----	۲-۱۲ کنترلرها
۲۵	-----	۲-۱۳ Application های تحت وب Apple :
۲۵	-----	۲-۱۴ محیط برنامه نویسی iPhone Apple :
۲۶	-----	۲-۱۵ زبان برنامه نویسی Phone :
۲۶	-----	۲-۱۵-۱ Net. بر روی iPhone :
۲۶	-----	۲-۱۵-۲ Flash بر روی iPhone :
۲۹	-----	۲-۱۶ سادگی رابط کاربری ios
۲۹	-----	۲-۱۶-۱ Android Honeycomb
۳۰	-----	۲-۱۶-۲ جذابیت Honeycomb
۳۱	-----	۲-۱۶-۳ عملکرد
۳۳	-----	۲-۱۷ تعداد نرم افزارهای موجود
۳۳	-----	۲-۱۸ محبوبیت
۳۷	-----	۲-۱۹ سؤال و جواب های اندروید
۳۸	-----	۲-۲۰ سؤال و جواب های آیفون
		فصل سوم: راه اندازی پروژه در xcode
۴۱	-----	۳-۱ راه اندازی پروژه در xcode
۴۳	-----	۳-۲ پنجره پروژه Xcode
۲		

۴۵	۳-۳ معرفی interface Builder (سازنده واسط)
۴۷	۳-۴ اضافه کردن یک لیبل به صفحه
۴۹	۳-۴-۱ توصیه
۴۹	۳-۵ تغییر مشخصه ها
۵۰	۳-۶ جلوه دادن به آیفون- پرداخت حرکات لمسی
۵۴	۳-۸ آوردن آن به صفحه home
	فصل چهارم: سرگرمی بیشتر با واسط کاربر
۵۶	۴-۱ سرگرمی بیشتر با واسط کاربر
۵۶	۴-۲ صفحه نمایشی پر از کنترل
۵۷	۴-۳ کنترل های فعال و غیر فعال
۵۹	۴-۴ ساخت برنامه کاربردی
۶۰	۴-۵ تعیین خروجی ها
۶۲	۴-۶ تعیین اکشن ها
۶۲	۴-۷ ساخت واسط (Interface)
۶۲	۴-۸ اضافه کردن نمای تصویر (image view)
۶۳	۴-۸-۱ سایز بندی کردن نمای تصویر
۶۴	۴-۹ ویژگی Mode
۶۴	۴-۱۰ اسلایدر Alpha
۶۵	۴-۱۰-۱ نادیده گرفتن پس زمینه
۶۵	۴-۱۱ ویژگی tag

۶۶	-----	۱۲-۴ چک باکس های طراحی
۶۷	-----	۱۳-۴ چک باکس های تعاملی
۶۸	-----	۱۴-۴ اضافه کردن فیلدهای متن
۶۹	-----	نتیجه گیری
۷۰	-----	مراجع:

چکیده

در این پروژه سعی شده است با یک زبان ساده نحوه برنامه نویسی و به دست گرفتن امکانات سخت افزاری چون سنسورهای حرکتی، مکان یاب، دوربین و ... را به علاقه مندان آموزش دهد و جهت استفاده از این پروژه نیاز به دانش قبلی برنامه نویسی نمی باشد، هر چند برنامه نویسان مخصوصاً برنامه نویسانی که با سیستکس زبان C آشنایی دارند در فهم این پروژه راحت تر از سایرین می باشند. لازم به ذکر است که هرچند این پروژه برای نسخه ۴ سیستم عامل ios طراحی شده است ولی پرواضح است، برنامه های طراحی شده با این نسخه قابل استفاده در دستگاه هایی که نسخه بالاتر نیز دارند مانند ios-5 می باشند، این سیستم عامل تا کنون تا ۷ نسخه ارائه شده است که نسخه ۷ آن به تازگی به بازار عرضه شده است.

هرچند محیط برنامه نویسی ios اپل مکیتاش می باشد ولی با نصب نسخه حک شده مکیتاش که به حکیتاش معروف می باشد می توان آن را روی کامپیوترهای غیر اپل نیز نصب نمود. جهت برنامه نویسی ios به ابزارهایی چون Xcode و SDK نیاز است.

مقدمه

این پروژه راهنمای خوبی است برای تمام کسانی که می خواهند قدم در این راه گذاشته و برنامه های کاربردی iOS را خودشان بسازند.

هدف ما آشنا کردن شما با دانش اولیه است. برای این که بدانید برنامه های کاربردی iOS چگونه کار می کنند و ساخته می شوند در طول مطالعه این پروژه برنامه های کاربردی کوچکی خواهید ساخت که هر کدام برای نشان دادن ویژگی های خاصی از iOS طراحی شده اند و به شما نشان می دهند که چگونه می توانید با این ویژگی ها و مشخصه ها کار کنید. در صورتی که به گروه ما ملحق شوید با کمک این پروژه و همچنین با خلاقیت اراده رهاگان را می یابید و با استفاده از مدارک و برنامه های کاملی که توسط اپل تولید شده است هر چیزی را که برای ساخت برنامه های کاربردی آی پد و آی تاچ حرفه ای تان نیاز دارید در اختیار خواهید داشت.

IOS نرم افزار اصلی تمام این برنامه ها بوده و همچنین جالب ترین نرم افزاری است که در طول این مدت عرضه شده است و دارای جالب ترین نرم افزارهای مربوط به موبایل می باشد به خصوص حالا که اپل یک سری لوازم شکیل و تکمیل شده برای ارتقاء برنامه های کاربردی iOS فراهم کرده و با انتشار بسته پیشرفته نسخه 4 نرم افزار ios (SDK) همه چیز بهتر خواهد شد.

آخرین نسخه ارائه شده توسط شرکت اپل iOS 7 می باشد.

تاریخچه

در ۱۰ ژانویه ۲۰۰۷ (۲۱ دی ۱۳۸۵)، اپل و استیو جابز، آیفون را معرفی کردند. محصولی که بزرگترین و گسترده‌ترین معرفی در زمان خود محسوب می‌شد. در طول ۶ سال بعد از آن، آیفون، آی‌پد و آی‌پاد تاچ به معنای واقعی، تعریف جهان از موبایل و کارکردهای آن را تغییر دادند. جهانی که به سرعت در حال پیشروی است و در آن، iOS، قدیمی‌ترین سیستم‌عامل در حال توسعه محسوب می‌شود. در حقیقت اپل از طریق ارتقاء و بهبودهای بی‌وقفه و سالانه، یکی از پرکاربردترین و پر امکاناتترین پلت‌فرم‌های بازار را با پشتیبانی بسیار عالی معرفی کرده است.

iOS، نرم‌افزاری است که به تلفن‌های همراه اپل جان بخشیده، سیستم‌عاملی که استفاده از آن برای اغلب کاربران، بسیار ساده است. پلت‌فرمی قدرتمند برای توسعه‌دهندگان اپلیکیشن که به کاربران خود در همه دستگاه‌ها تجربه‌ای مشابه ارائه می‌کند. احتمالاً قابل توجه‌ترین بخش iOS فعلی، شباهت بسیار زیاد آن به نسخه اولیه iOS در سال اول است. با این حال، اپل، انبوهی از امکانات و ویژگی‌های جدید را از آن زمان ارائه نموده است. اپل، به دور از خطر یکنواخت شدن ویژگی‌های سیستم‌عاملش به مرور زمان، iOS را به گونه‌ای راهبری نمود که همچنان سرزنده و جذاب بوده و ویژگی‌های اختصاصی بهتری نسبت به هریک از سیستم‌عامل‌های موجود در خود داشته باشد iOS ۶، از این مرحله نیز فراتر رفته است. چگونه از سیستم‌عاملی ساده و بدون پشتیبانی از اپلیکیشن‌های Third Party، چند وظیفگی و یا حتی Copy/Paste، در سال ۲۰۰۷، به سیستم‌عامل امروزی رسیدیم.

سیستم عامل موبایل اپل، محصولی از قلب OS X

زمانی که آیفون اول معرفی شد، اپل اعلام کرد که iOS، مانند OS X، از هسته یونیکس متولد شده و از ابزارهای مشابهی استفاده می‌کند. با همه اینها، این مسأله کاملاً مشهود بود که علیرغم اشتراکات بسیار زیادی که میان OS X و iOS وجود دارد، iOS برای اینکه بتواند یک سیستم‌عامل جداگانه محسوب شود، به اندازه کافی با OS X متفاوت بود.

هنگامی که آیفون عرضه شد، iOS را iPhone OS می‌نامیدند و این نام برای سه سال باقی ماند و تنها پس از معرفی iOS ۴ در ژوئن ۲۰۱۰ بود که نام آن به iOS تغییر یافت، با وجود این در ادامه متن و برای راحتی، ما همه نسخه‌های سیستم‌عامل اپل را iOS می‌نامیم.

تولد آیفون iOS

اگرچه ممکن است تصور این مطلب کمی سخت باشد، ولی زمان معرفی آیفون اول، سیستم‌عامل آن از همه رقبا عقبتر بود و در مقایسه با امکانات سایر سیستم‌عامل‌ها حرف زیادی برای گفتن نداشت.

در سال ۲۰۰۷ ویندوز موبایل، پالم، سیمبین و بلک‌بری با داشتن مجموعه کاملی از ویژگی‌های متنوع، سیستم‌عامل‌های موفق و قدرتمندی محسوب می‌شدند. در مقابل آنها، iOS محصولی بود که از G۳، چندوظیفگی، اپلیکیشن‌های Third Party و حتی از Copy/paste پشتیبانی نمی‌کرد. اول، از ضمیمه کردن فایل‌ها به ایمیل، از پیام‌های چندرسانه‌ای (MMS)، از اپلیکیشن‌های دریافت و ارسال ایمیل، از شخصی‌سازی صفحه اصلی، پشتیبانی نمی‌کرد. در آن زمان، iOS، فایل‌های سیستمی را از دید کاربر مخفی می‌کرد و امکان ویرایش فایل‌های متنی را نداشت. از شماره‌گیری صوتی پشتیبانی نمی‌کرد و برای توسعه‌دهندگان و هکرها به طور کامل بسته و غیرقابل دسترسی بود. اپل در آن هنگام، به جای رقابت در زمینه سخت‌افزار، بر روی ایجاد یک تجربه کاربری بی‌نظیر تمرکز نمود.

با همه اینها، تمام امکاناتی که در آن زمان در iOS وجود نداشتند، حالا در دسترس و توسط اغلب کاربران شناخته شده هستند. اپل در آن هنگام، به جای رقابت در زمینه سخت‌افزار، بر روی ایجاد یک تجربه کاربری بی‌نظیر تمرکز نمود. برای همین در سال ۲۰۰۷، بر روی سرعت، سازگاری اپلیکیشن‌ها با یکدیگر و ارائه چندین ویژگی بسیار بهتر نسبت به رقبا تمرکز نمود. البته، iOS در آن زمان هم خلاقیت‌ها و نوآوری خیره‌کننده‌ای در خود داشت که در ادامه به سه مورد از آنها، که خلاقیت‌هایی انقلابی در صنعت موبایل محسوب می‌شدند اشاره می‌کنیم

فصل اول

سیستم عامل و انواع آن

۱-۱ مقدمه ای بر ایجاد سیستم عامل ها

در فرهنگ رایانه سیستم عامل نرم افزاری است که مدیریت منابع رایانه را به عهده گرفته و بستری را فراهم می سازد که برنامه های کاربردی اجرا شده و از خدمات آن استفاده کنند .

۱-۱-۱ سیستم عامل چه کاری انجام می دهد.

یک سیستم کامپیوتری پیشرفته از یک یا چند پردازنده، مقداری حافظه اصلی، دیسک ها، چاپگرها، صفحه کلید، صفحه نمایش، واسط های شبکه ای و دیگر دستگاه های ورودی و خروجی تشکیل شده است. اگر سیستم یکپارچه ای برای مدیریت این منابع وجود نداشته باشد، هر برنامه باید به تنهایی این کار را انجام دهد. سیستم عامل یک لایه نرم افزاری فراهم می کند که وظیفه مدیریت منابع سیستم را از دوش برنامه های کاربردی رها کرده و کار برنامه نویسی را ساده تر می نماید.

۱-۲ انواع سیستم عامل :

سیستم عامل تک پردازنده : این نوع سیستم عامل ها ، سیستم عامل های نسل چهارم (نسل فعلی) هستند که بر روی یک پردازنده اجرا می شوند.

سیستم عامل شبکه ای : این نوع سیستم عامل ها ، از کنترل کننده های واسط شبکه و نرم افزار های سطح پایین به عنوان گرداننده استفاده می کنند و برنامه هایی برای ورود به سیستم های راه دور و دسترسی به فایل از راه دور در آنها به کار گرفته می شود

سیستم عامل توزیع شده : این سیستم عامل ها خود را مانند سیستم عامل های تک پردازنده به کاربر معرفی می کنند ، اما در عمل از چندین پردازنده استفاده می کنند. این نوع سیستم عامل در یک محیط شبکه ای اجرا می شود در این نوع سیستم یک برنامه پس از اجرا در کامپوترهای مختلف جواب نهایی به سیستم اصلی کاربر بر می گردد سرعت پردازش در این نوع سیستم بسیار بالاست.

سیستم عامل بی درنگ : از این نوع سیستم های عامل برای کنترل ماشین آلات صنعتی، تجهیزات علمی و سیستم های صنعتی استفاده می گردد. یک سیستم عامل بلادرنگ دارای امکانات محدود در

رابطه با بخش رابط کاربر و برنامه‌های کاربردی مختص کاربران هستند. یکی از بخش‌های مهم این نوع سیستم‌های عامل، مدیریت منابع موجود کامپیوتری بگونه‌ای است که یک عملیات خاص در زمانی که می‌بایست، اجراء خواهند شد، مهم‌تر اینکه مدیریت منابع بگونه ایست که این عمل در هر بار وقوع، مقدار زمان یکسانی بگیرد.

۳-۱ وظایف سیستم عامل چیست ؟

سیستم عامل دو کار عمده انجام می‌دهد:

در نگرش پایین به بالا، منابع منطقی (مانند فایل‌ها) و منابع فیزیکی (مانند دستگاه‌های سخت افزاری) رایانه را مدیریت و کنترل می‌کند.

در نگرش بالا به پایین، وظیفه سیستم عامل این است که یک ماشین توسعه یافته (Extended Machine) یا ماشین مجازی (Virtual Machine) را به کاربران ارائه کند تا آنها بتوانند آسان‌تر برنامه نویسی نمایند و درگیر پیچیدگی‌های سخت افزاری رایانه نشوند.

به طور کلی، وظایف سیستم عامل شامل موارد زیر است:

- استفاده بهینه از منابع و جلوگیری از به هدر رفتن آنها
- تخصیص و آزاد سازی منابع
- اداره صف‌ها و زمان بندی استفاده از منابع
- حساب داری (Accounting) میزان استفاده از منابع
- ایجاد امنیت (security)
- ایجاد، حذف و اداره فرایندها
- ایجاد مکانیسم‌های ارتباط بین فرایندها و همگام سازی آنها
- مدیریت فایل‌ها و پوشه‌ها
- مدیریت حافظه‌های اصلی و جانبی

- برقراری امکان دسترسی چندتایی (Multiaccess) و اجرای هم رونند (Concurrent) فرایند ها
- به اشتراک گذاری منابع (Resource Sharing)
- تعیین راهکار هایی برای اداره بن بست (deadlock) ها
- جلوگیری از شرایط رقابتی (Race Condition) و تداخل یا در هم قفل شدن (Interlock) فرایند ها
- جلوگیری از گرسنگی (Starvation)

انتخاب یک سیستم عامل یک نیاز شخصی است و در مقایسه سیستم ها با هم سعی شده نقاط ضعف و قدرت آنها مطرح شود و بحث ها به طور جانبه دارانه از یک سیستم عامل خاص نباشد .

در آخر باز هم متذکر می شوم که انتخاب یک سیستم عامل بستگی به نیاز هر شخص و کار انجامی ان بستگی دارد پس همه ی سیستم عامل ها برای همه ی کارها نوشته نشده اند و هر کار نیاز به سیستم عامل و سخت افزار مورد نیاز خود دارد .

روند رو به گسترش تولید سیستم های عامل جدید و هوشمند برای گوشی های تلفن همراه باعث تبدیل موبایل از یک وسیله ارتباطی ساده به یک رایانه جیبی با کاربرد های متنوع شده است .

طی سال های اخیر با افزودن قابلیت دسترسی به اینترنت در گوشی های تلفن های همراه ، رقابت برای ساخت سیستم های عامل جدید و هوشمند برای موبایل ها نیز گسترش یافته است . اکنون گوشی های تلفن همراه با استفاده از سخت افزار های پیچیده و نرم افزار های پیشرفته گوناگون وارد عصر جدیدی از حیات خود شده اند . اولین بار که تلفن همراه به بازار آمد و ارتباط افراد در تمامی نقاط بدون استفاده از برق و خطوط کابلی عادی تلفن ی برقرار شد ، بسیاری دچار شگفتی شدند و کمتر کسی تصور می کرد روزی موبایل ها قابلیت تصویربرداری و عکسبرداری پیدا کنند .

در ابتدا از تلفن همراه تنها برای ارتباط عادی استفاده می شد اما با گذر زمان قابلیت های دیگری نیز به این دستگاه ارتباطی افزوده شد و پس از چند سال دیگر برقراری ارتباط آنالوگ کلامی تنها ویژگی

موبایل نبود و دارندگان این دستگاه می توانستند با استفاده از دوربینی که در گوشی های تلفن همراه تعبیه شده بود عکس بگیرند و بعدها قابلیت فیلمبرداری نیز به آن افزوده شد.

به موازات این تغییرات گوشی های تلفن همراه صاحب قابلیت اجرا و ارسال پیام های کتبی کوتاه موسوم به اس ام اس (پیامک) نیز شدند و به مرور نیز ارسال تصاویر با اینفرارد و سپس با بلوتوث از طریق یک موبایل به موبایلی دیگر فراهم شد. اما با گسترش روزافزون فناوری های رایانه ای و فناوری اطلاعات و کاربرد اینترنت در زندگی افراد، سازندگان گوشی های تلفن همراه و سایر شرکت های بزرگ ساخت نرم افزار های کامپیوتری مانند مایکروسافت به فکر ساخت و طراحی سیستم های عامل برای موبایل افتادند.

در پی این تصمیمات، شرکت های مطرح ساخت گوشی های تلفن همراه نسل های دوم و سوم گوشی های خود را عرضه کردند. این گوشی ها دارای امکانات سخت افزاری بیشتری بودند که شامل پردازنده، حافظه و دیسک سخت بود و تقریبا به صورت یک رایانه کوچک طراحی شده بود.

در این زمان بود که این شرکت ها به اهمیت نرم افزار در این گوشی ها پی برده و رقابت نرم افزاری در کنار رقابت سخت افزاری شروع شد. مایکروسافت یکی از شرکت های ی بود که قبلا روی سیستم عامل رایانه های جیبی خود کار کرده و ویندوز سی ای CE را به همین منظور طراحی کرده بود. این ویندوز قابلیت این را داشت که روی موبایل ها نیز نصب شود.

بعد از رسمیت یافتن تلفن های همراه نسل جدید، شرکت های مطرح در این صنعت که به مشکلات بی شمار سیستم عامل سی ای پی برده بودند و نمی خواستند نرم افزار این صنعت را در انحصار یک شرکت نگه دارند با همکاری هم، سیستم عامل جدیدی برای موبایل ها به وجود آوردند که سیمبیان نام گرفت. البته رقبای دیگری نیز مثل سیستم عامل لینوکس در این صنعت وجود دارند. در اصطلاح به موبایل های ی که از این سیستم عامل ها استفاده می کنند تلفن های همراه هوشمند (SmartPhone) گفته می شود.

۴-۱ گوشی های بدون سیستم عامل ، گوشی های جاوا:

تا همین چند سال اخیر واژه سیستم عامل در بین گوشی های تلفن همراه معنا و مفهوم خاصی نداشت . یعنی گوشی های تلفن همراه از سیستم عامل خاصی استفاده نمی کردند و شاید صحیح تر بتوان گفت سیستم عاملی برای گوشی های تلفن همراه وجود نداشت ! اما به مرور با پیشرفت سرسام آور دنیای فن آوری موبایل سیستم های موبایلی که قبلا وجود داشت نام گرفتند . یعنی اینکه با به وجود آمدن سیستم عامل های مختلف واژه ای به نام گوشی های بدون سیستم عامل بوجود آمد.

این نوع از گوشی ها از نظر ساختار نرم افزار جزو ساده ترین نوع گوشی های تلفن همراه هستند . به صورت غیر تخصصی می توان آنها را در سه گروه زیر دسته بندی کرد:

- گوشی های ساده بدون سیستم عامل که قابلیت ساپورت نرم افزار ندارند ، از امکاناتی چون دوربین و ... نیز برخوردار نیستند : نوکیا ۱۲۰۰ ، ۱۱۰۰ ، و ...
- گوشی های ساده بدون سیستم عامل که قابلیت ساپورت نرم افزار های مخصوص تولیدی فقط توسط سازنده را دارند ، گاهی از دوربین و ... نیز در سطح پایین برخوردار هستند : نوکیا ۶۰۲۰ و تعدادی از گوشی های سامسونگ و موتورولا و ...
- گوشی های بدون سیستم عامل که قابلیت نصب تم های مخصوص ، بازی ها و نرم افزار های جاوا را دارند ، دارای دوربین عکاسی گاهی با کیفیت بسیار بالا نیز هستند . در بین عموم مردم به گوشی های جاوا معروف هستند. مانند اکثر گوشی های سونی اریکسون : w770 , k750 , w810 نوکیا : ۶۵۰۰ , ۵۳۱۰ و ...

سه نوع بالا اگر چه در دسته بندی گوشی های بدون سیستم عامل قرار دارند (سیستم دارند، در واقع سیستم ها کاربری ساده ویژه کارخانه برای گوشی) اما مسلم است نوع سوم تفاوت ها و گستردگی های بسیار زیادی را با دو نوع اول دارند و امروزه نیز قسمت بزرگی از بازار را تشکیل می دهند و در واقع گوشی های با کاربری فقط ضروری نام گرفته اند.

۱-۵ انواع سیستم عامل های موبایل :

سیستم عامل سیمبین (Symbian OS)

سیستم عامل لینوکس (LinuX)

سیستم عامل پالم (paLm)

سیستم عامل بلک بری (BlaCk Berry)

سیستم عامل ویندوز موبایل (WindoS MOBiLe)

سیستم عامل مک آی او اس (اپل) (appLe MaC ios)

سیستم عامل آندروید گوگل (AndrOid)

سیستم عامل بادا (Bada)

سیستم عامل میامو (MaemO)

سیستم عامل میگو (MeeGo)

فصل دوم

آشنایی با سیستم عامل iOS

IOS ۲-۱

ios به انگلیسی ios: در گذشته iPhone OS یک سیستم عامل تلفن همراه است که در ابتدا برای آی فون و آی پاد تاج توسعه داده می شد، از آن زمان به بعد برای استفاده در سایر دستگاه های شرکت اپل مانند آی پد و اپل تی وی گسترش یافت. شرکت اپل مجوز استفاده از آی او اس برای نصب بر روی سخت افزارهای شخص ثالث را نمی دهد. اپ استور در ۶ مارس ۲۰۱۲ دارای بیش از ۵۵۰,۰۰۰ اپلیکیشن بود که جمعاً بیش از ۲۵ بیلیون بار دانلود شده اند. در سه ماهه آخر سال ۲۰۱۰ این سیستم عامل ۱۶٪ از سهم سیستم عامل تلفن های هوشمند را در اختیار داشت، پایین تر از اندروید گوگل و سیمبین نوکیا. این سیستم عامل در ماه مه ۲۰۱۰ در ایالات متحده آمریکا، ۵۹٪ از مصرف داده های وب تلفن همراه (از جمله استفاده از آی پاد تاج و آی پد) را به خود اختصاص داد.

رابط کاربری آی او اس بر اساس مفهوم دستکاری مستقیم (concept of direct manipulation)، با استفاده از حرکات چند لمسی (multi-touch gestures) طراحی شده است. عناصر رابط کنترل از switches: buttons و sliders تشکیل شده است. پاسخ به ورودی کاربر فوری است و رابط سیال را فراهم می کند. تعامل با سیستم عامل حرکاتی را شامل می شود از این جمله حرکات می توان: swipe, tap, pinch, reverse pinch که هر کدام از اینها تعاریف خاصی را در چارچوب IOS سیستم عامل و رابط کاربری چند لمسی آن را دارا می باشند... شتاب سنج داخلی توسط برخی از برنامه های کاربردی مورد استفاده قرار گیرد که برای پاسخ به تکان دادن دستگاه (یکی از نتایج شایع ان دستور خنثیسازی است) و یا چرخش آن را به سه بعد (یکی از نتایج شایع ان تعویض از عمودی به افقی است).

IOS mode از OX مک مشتق شده است، که با آن که سهام آن به بنیاد داروین تعلق دارد، و در نتیجه این سیستم عامل شبه یونیکس از طبیعت است.

در IOS، چهار انتزاع لایه وجود دارد: لایه هسته سیستم عامل، لایه هسته خدمات، لایه رسانه ها، و کاکائو لایه لمسی است. نسخه کنونی سیستم عامل ۱,۱,۵ با استفاده از حدود ۸۶۰ مگابایت ذخیره سازی دستگاه که با مدل های دیگر متفاوت است

۲-۲ ویژگی‌ها

۲-۲-۱ صفحه اصلی

صفحه اصلی (ارائه شده توسط (" SpringBoard ")) نمایش آیکون‌های نرم افزار و بارانداز در پایین صفحه نمایش که در آن کاربران می‌توانند برنامه‌هایی را که اغلب استفاده می‌کنند پین کنند. صفحه اصلی هر زمان که کاربر بر روی دستگاه سوئیچ کند و یا دکمه "صفحه اصلی" را فشار دهد (دکمه فیزیکی بر روی دستگاه) نمایان می‌شود. صفحه نمایش دارای یک نوار وضعیت در بالا برای نشان دادن داده‌ها: مانند زمان، میزان مصرف باتری و قدرت سیگنال است. بقیه صفحه نمایش به برنامه در حال اجرا اختصاص داده شده است. از نسخه IOS ۳.۰ تا به حال، تابع Spotlight Search function در صفحه سمت چپ صفحه اصلی روی صفحه نمایش در دسترس شده است. این تابع به کاربران اجازه می‌دهد تا درون رسانه‌ها، برنامه‌ها، ایمیل‌ها، دفترچه تلفن‌ها و فایل‌های مشابه را جستجو کنند.

۲-۳ مرکز هشدار دهنده (Notification Center)

در بروز رسانی IOS ۵، قابلیت اطلاعیه‌ها به طور کامل دوباره طراحی شده است. اطلاعیه‌ها در حال حاضر در یک پنجره است که می‌تواند از بالای صفحه نمایش کشیده شود و اگر اطلاعیه دریافتی را لمس کنید، شما به برنامه‌ای که ارسال اطلاع رسانی کرده وارد می‌شوید.

۲-۴ پوشه‌ها (Folders)

IOS با معرفی یک سیستم ساده برای پوشه‌ها آمد. هنگامی که برنامه‌های کاربردی در حالت "jiggle mode" هستند، هر دو برنامه را می‌توان در بالای هر یک از دیگری برای ایجاد یک پوشه کشیده، و از آن پس، برنامه‌های بیشتری را می‌توان با استفاده از همین روش به پوشه اضافه کرد، تا ۱۲ برنامه بر روی آی فون و iPod Touch و ۲۰ برنامه بر روی iPad.

۲-۵ عملکرد چند تکلیفی (Multitasking)

قبل از iOS 4، Multitasking فقط برای بعضی از برنامه‌های موجود در دستگاه قابل اجرا بود. با شروع iOS 4 در نسل سوم و جدیدتر دستگاه‌های iOS، multitasking از طریق هفت پس زمینه APIها پشتیبانی شد:

۱. برنامه در پس زمینه ادامه پیدا می‌کند تا زمانی که پخش صوتی و یا تصویری اجرا می‌شود.

۲. برنامه به صورت معلق می‌باشد هنگامی که یک تماس تلفنی در حال انجام است.

۳. نرم افزار اطلاع از تغییر محلها

۴. فشار اطلاعیه‌ها.

۵. اطلاعیه محلی. برنامه زمانبندی اطلاعیه محلی که در یک زمان از پیش تعیین شده تحویل داده شده‌است.

۶. اتمام کار. برنامه از سیستم درخواست زمان اضافی برای تکمیل یک کار خاص می‌کند

۷. برنامه هیچ کد اجرا نمی‌کند و ممکن است از حافظه در هر زمان حذف شده‌است

۲-۶ راه گزینی برنامه‌های کاربردی (Switching applications)

بادوبار کلیک کردن بر روی دکمه Home. Switching applications را فعال می‌سازد. که یک رونده حوض مانند از پایین، با حرکت دادن محتویات صفحه نمایش تا بالای صفحه نمایش ظاهر می‌شود. انتخاب یک آیکون سوئیچ به برنامه سمت چپ آیکون که به عنوان کنترل‌های موسیقی، و قفل چرخش عمل می‌کنند. نگه داشتن آیکون‌ها به طور خلاصه آنها را به "جنش" در آورده (به همین ترتیب به صفحه اصلی) و اجازه می‌دهد تا کاربر برنامه‌ها را به بسته شدن وادار کند.

۲-۷ مرکز بازی (Game Center)

مرکز بازی یک شبکه اجتماعی بازی‌های چند نفره آنلاین منتشر شده توسط اپل است که این اجازه را به کاربران می‌دهد تا "دوستان را به بازی دعوت کنند، یک بازی چند نفره از طریق matchmaking شروع کنند. به پیگیری امتیازهای خود، و مقایسه آنها با امتیازهای دوستان خود در leader board بپردازند. مرکز بازی یک شبکه اجتماعی بازی‌های چند نفره آنلاین منتشر شده توسط اپل است که این اجازه را به کاربران می‌دهد تا "دوستان را به بازی دعوت کنند، یک بازی چند نفره از طریق matchmaking شروع کنند. به پیگیری امتیازهای خود، و مقایسه آنها با امتیازهای دوستان خود در leader board بپردازند

۲-۸ محیط برنامه نویسی

برنامه‌ها حتما باید به طور خاص برای IOS و معماری ARM نوشته و کمپایلد شوند. مرورگر وب سفاری برنامه‌های کاربردی وب را با دیگر مرورگرهای وب پشتیبانی می‌کند. برنامه‌های بومی مجاز شخص ثالث برای دستگاه‌های در حال اجرا IOS ۲.۰ و بعد از طریق فروشگاه app اپل در دسترس هستند.

در هفدهم اکتبر سال ۲۰۰۷ برنامه ای سرگشاه به "خبرهای داغ" و بلاگ شرکت اپل "apple" استیو جابز اعلام کرد که کیت نرم افزاری توسعه یافته sdk در سومین دسته ی توسعه دهنده ها در فوریه ۲۰۰۸ قابل دسترسی خواهد بود sdk . ماوس ۲۰۰۸ منتشر شد و به توسعه دهندگان اجازه داده شد تا برای آیفون و آی پاد لمسی استفاده شود و همچنین در شبیه ساز آیفون تست شود. با این وجود بارگیری یک کاربرد برای دستگاه ها فقط بعد از پرداخت حقوق توسعه دهنده برنامه ی آیفون امکان پذیر خواهد بود . دستمزد ها برای پیوستن به برنامه ها مربوط به ios و OSX برای هر گواهینامه توسعه دهنده ها ۹۹۰۰ دلار تعیین شده است. این دستمزد ۹۹۰۰ دلاری باید سالانه به طور منظم برای هر توسعه دهنده پرداخت شود تا گواهینامه آنها حفظ شود . از بیستم جولای ۲۰۱۰ شرکت اپل "apple" منتشر کرد که داندلود x code در فروشگاه هک اپل برای تمامی استفاده کنندگان ioslion رایگان می باشد. کاربران با استفاده از نسخه ی رایگان x code می توانند کاربر های ios

رایجاد کنند و با این وجود توسعه دهنده ها نمی توانند کاربر هایشان را روی یک وسیله مادی ios امتحان کنند. آنها را به فروشگاه اپ "app" می فرستند یا از کاربر هایشان بدون پرداخت ۹۹۰۰ دلاری اولیه برای توسعه دهنده iphone یا دستمزد توسعه دهنده ی برنامه ها سود می برند.

از زمان انتشار x code , x code ۳.۱ محیط توسعه یافته ای برای ios sdk بوده است . کاربرهای iphone مانند ios و osx در objective نوشته شده است.

توسعه دهندگان قادر اند هر قیمتی روی حداقل تعیین شده برای کاربرد هایشان که فروشگاه ها "app" طبقه بندی می شوند تعیین کننده ۳۰ درصد ان را شرکت اپل خواهد گرفت و ۷۰ درصد دیگر برای توسعه دهنده می رود . به طور متناوب آنها ممکن است تصمیم به انتشار رایگان کاربر بگیرند که به پرداخت هیچ هزینه ای برای انتشار یا توزیع کاربر به جز حق عضویت احتیاج نیست.

بعد از اولین انتشار IOS مطالعه ای برای تنوع های مختلف مراکز هک در مورد افزودن عملکرد هایی که به شرکت اپل اجازه داده نمی شد انجام گرفت. قبل از پیش نمایی IOS بومی در سال ۲۰۰۸ در فروشگاه اپ "app" محرک اصلی برای نصب زندان گریز در کار برد های بومی شکل گرفته بود که تا ان زمان به شرکت اپل اجازه نصب داده نشده بود.

شرکت اپل ادعا می کرد که طراحی نرم افزار به روز شده IOS را منتشر نخواهد کرد مخصوصا برای شکستن این ابزار ها (مگر کاربر هایی که برای قفل گشایی سیم کارت اجرا می شود. با این وجود با هر به روز رسانی ios زندان گریز سر هم بندی نشده قبلی معمولا سرهم بندی می شد و به کار برده می شد.

از زمان آمدن ios بومی اپل در فروشگاه "app" اپ " و همراه ان سومین قسمت کار بردی انگیزه کل برای زندان گریز تغییر پیدا کرده است . مردم به دلایل زیادی از قبیل دسترسی به پوشه های سیستم نصب تم ابزار های نمایشی و تعدیل spring board از زندان گریز استفاده می کند در ابزار زندان گریز این امکان را فراهم می کند از قبیل "اندروید" و linux kernel . در ابتدا کاربران به خاطر محدودیت های ios ابزار هایشان را زندان گریز می کردند.

در سال ۲۰۱۰ eff با موفقیت دفتر حقوق چاپ ایالات متحده را مجاب کرد که اجازه ی معافیت برای ممنوعیت کلی برای احتراز از سیستم های حفاظتی زیر نظر dmca را گرفت. این اجازه معافیت زندان گریز آیفون حصول تنها هدف برای کاربرد های قانونی است که به آیفون اضافه شده است . این معافیت تاثیری روی روابط قراردادی بین شرکت اپل و مالک آیفون ندارد. برای مثال ابطال گارانتی زندان گریز آیفون با این وجود این به تنهایی وابسته یه صلاحدید شرکت اپل است که آیا که ابزار های زندان گریز را روی ابزاری که احتیاج به تعمیر دارند نصب خواهند کرد . قابل به ذکر است که بسته به روش های استفاده شده تاثیرات زندان گریز پایدار بوده و می تواند به حالت کلی برگردانده شود. در چنین زمانی دفتر حق چاپ قفل گشایی آیفون را از معافیت ض احتراز dmca معاف کرد . قفل گشای آیفون به آیفون اجازه میداد که قابل استفاده با هر حامل بی سیم باشد که تکنولوژی یکسان gsm یا cdma یا هر مدل تلفنی مخصوص که برای بهره برداری طراح شده است.

۹-۲ UNLOCKING

قفل گشایی موضوع اصلی قفل سیم کارت در ابتدا خیلی از حامل های بی سیم اجازه قفل گشایی آیفون برای استفاده دیگر حامل ها را به مالک آیفون نمی دادند. موبیلتی at,t به مالک آیفون اجازه داد تا در شرایط قرارداد شان برای قفل گشایی آیفون رضایت داشته باشد. دستورالعمل ها برای قفل گشایی ابزار از سوی شرکت اپل امکانپذیر میباشد. در این صورت استفاده منبع AT,T آیفون روی دیگر شبکه های GSM از قبیل t موبایل و دیگران اجازه داده شد.

با این وجود چون t موبایل قبلا گروه دیگری غیر از AT,T برای سیگنال های اطلاعاتی g3 استفاده میکرده است آیفون در سرعت ۱۹۰۰ مگاهرتز g3 شبکه T-mobile فعال خواهد بود.

۱۰-۲ Digital right management

مدیریت حقوق دیجیتال : توافق و ماهیت مالکیت ios انتقادات دیگری را ایجاد کرد مخصوصا مدافعان حقوق دیجیتال از قبیل eff(بنیان الکترونی مرزی) مهندس کامپیوتر و فعال محیط زیست

بروستر کاهل متخصص قوانین اینترنت جانتون زیتون و بنیان نرم افزار آزاد fsf که به رویداد مقدماتی آی پد و اهداف آی پد برای فعالیت های سازمان یافته ب طراحی معیوب اعتراض کرد . میکروسافت رقیب نیز از طریق سخنگوی روابط عمومی تاثیر خط و مشی شرکت اپل را نفی کرد . محدودیت های تحمیل شده ی مرد بحث به طراحی به طراحی ios یعنی مدیریت حقوق دیجیتال (drm) قصد دارد رسانه خریداری شده پلت فرم شرکت اپل را قفل کند. مدل توسعه یافته (حق اشتراک سالانه ای) برای کاربرد های توسعه یافته ی IOS تعمیم داده شده است . تمرکز برای تصویب فرآیند کاربرد ها، کنترل کلی شرکت اپل و ممنوعیت های خود پلت فرم مخصوصا توانایی شرکت اپل برای سلب صلاحیت اپل یا حذف کاربرد های آن مورد بحث خواهد بود . بعضی در اشتراک تکنولوژی موضوعی بیان کرده اند که ممنوعیت IOS رویه ی رو به رشدی را برای دستیابی شرکت اپل به محاسبه را داشته است . مخصوصا تغییر پیوسته ی شرکت اپل از دستگا هایی که افراد میتوانند آن را تعمیر کنند و با توجه به پتانسیل این قبیل ممنوعیت ها برای فرو نشاندن نو آوری های نرم افزاری است . با این وجود به غیر از اپل کسانی هستند که حمایتشان را از مدل محرمانه ابراز کردند . توسعه دهنده ی فیس بوک جو هویت که سابق براین علیه شرکت اپل به سخت افزاری به عنوان نمونه ای وحشتناک اشاره داشته است متعاقبا استدلال کرده است که قفل کاربرد ها در " I pad" آی پد با قفل کاربرد های آن.

ایده اصلی این سیستم عامل استفاده از رابط کنترلی چند لمسی است. برای مثال برای جابجا کردن صفحات در ios تنها نیاز به کشیدن یک انگشت روی صفحه است. با قرار دادن دو انگشت روی صفحه و دور کردن آنها از هم نیز می توان بر روی صفحات وب و عکس ها زوم کرد ios. از نظر معماری از چهار لایه Core OS ، Core Services ، Media Layer و Cocoa Touch تشکیل شده است. با انتشار نسخه شماره 4 Apple قابلیت Multi-Tasking را به سیستم عامل ios اضافه کرد. نسخه 5 نیز اولین سری از سیستم عامل ios بر پایه پردازش ابری است.

ios از نظر معماری و کاربردی شباهت زیادی به سیستم عامل Mac OS دارد. در ios همانند Mac OS از مرورگر Safari برای مرور وب و از نرم افزار iTunes برای پخش موزیک و مدیریت کتاب ها، دفترچه تلفن و Application استفاده می شود. یکی از موفقیت های Apple در

ساخت این سیستم عامل، راه اندازی فروشگاه **App Store** است. این فروشگاه تا پایان ماه مه ۲۰۱۱ دارای بیش از ۵۰۰ هزار **Application** بوده است. شرکت **Apple** نرم افزارهای **App Store** را با بستن قرارداد با برنامه نویسان منتشر می کند و هر کس می تواند با استفاده از ابزار **SDK** سیستم عامل **iOS**، **Application** طراحی کند و بعد از دریافت مجوز از **Apple** آن را به قیمت دلخواه در **App Store** به فروش رساند. سیستم عامل **iOS** از ابتدای انتشار مورد هدف هکهای سیستم عامل قرار گرفت. به این ترتیب هکرها موفق شدند هم قفل شبکه موبایل (**Sim Lock**) گوشی **iPhone** را بشکنند و هم محدودیت نصب **Application** ها بر روی این سیستم عامل را با استفاده از رخنه های امنیتی در مرورگر **Safari** و الگوریتم خواندن فایل های **PDF** دور بزنند. به این کار اصطلاحاً **Jailbreak** گفته می شود.

۱۱-۲ معماری سیستم عامل **Phone**:

سیستم عامل **iPhone** و سیستم عامل **Mac** از نظر معماری بسیار شبیه یکدیگر هستند و تنها تفاوتشان این است که لایه ای بالایی در **iPhone**، **Cocoa Touch** است و در **Mac**، **Cocoa Framework** لایه پایینی **Core OS** می باشد که بنیاد سیستم عامل است. وظیفه ای این لایه مدیریت حافظه، فایل سیستم، شبکه و کارهای دیگر سیستم عامل می باشد. این لایه مستقیماً با سخت افزار در ارتباط است **Core OS**. از کامپوننت های نظیر زیر تشکیل شده است:

کرنل **OS X**، **Mach ۳.۰**، **BSD**، **Socket** ها، امنیت، مدیریت نیرو، **Keychain**، **Certificate** ها، فایل سیستم، **Bonjour** لایه **Core Services** یک تجرید را بر روی سرویس هایی که توسط لایه **Core OS** فراهم می شوند ارائه می دهد. این لایه دسترسی بنیادی به سرویس های سیستم عامل **iPhone** را فراهم کرده و از کامپوننت های زیر تشکیل شده است

Collection ها، **Address Book**، شبکه، دسترسی فایل، **SQLite**، **Core Location**، سرویس های **Net**، **Threading**، **Preferences**، ابزارهای **URL** لایه **Media** سرویس های مالتی مدیا را فراهم می کند که می توانید در برنامه های **iPhone** و **iPad** استفاده نمایید و از کامپوننت های زیر تشکیل شده است:

،Video Playback ،Audio Recording ،Audio Mixing ،OpenGL ،Core Audio
Cocoa ،OpenGL ES ،Core Animation ،Quartz ،PDF ،TIFF ،PNG ،JPG
Touch یک لایه ی تجرید فراهم کرده تا کتابخانه های مختلفی را برای برنامه نویسی iPhone و
iPad نظیر مثال های زیر ارائه دهد:

View ،Accelerometer ، Multi-Touch های کنترل های ، Multi-Touch های event
،Image Picker ،People Picker ،Web View ها، Alert ،Localization ،Hierarchy

۲-۱۲ کنترلرها

از لحاظ سخت افزاری iPhone بجای پردازنده های X ۸۶ از پردازنده های ARM استفاده می کند.
این پردازنده ۳۲ بیتی است و به دلیل سادگی و سبک بودن در دستگاه های موبایل، کنسول های بازی
همراه، دستگاه های پخش موسیقی، ماشین حساب ها و ... کاربرد دارد.

۲-۱۳ Application های تحت وب Apple :

در سال ۲۰۰۷ امکان نوشتن برنامه های تحت وب و اجرای آنها بر روی iPhone را با استفاده از
مرورگر Safari فراهم کرد. با توجه به دشوار بودن برنامه نویسی برای iPhone ، این امکان برای
طراحان بسیار جذاب بود . زیرا براحتی می توانستند تکنولوژی هایی مانند Ajax را به iPhone
بیاورند. اما قوی نبودن موتور JavaScript مرورگر Safari باعث شد تا برنامه های تحت وب
iPhone چندان دلچسب نباشند و طرفداران زیادی نداشته باشند.

۲-۱۴ محیط برنامه نویسی iPhone Apple :

در سال ۲۰۰۸ یک IDE برای iPhone ارائه کرد. این SDK محیط برنامه نویسی - در Xcode
اجرا می شود. Xcode همان محیط برنامه نویسی Mac OSX است. این محیط یک شبیه ساز
iPhone نیز برای تست برنامه های نوشته شده دارد. اما برای ایجاد فایل های که روی دستگاه iPhone
اجرا شود حتما باید یک نسخه کامل از این IDE را خریداری کرده باشید. برنامه نویسان می توانند

برنامه‌های خود را روی APP Store قرار دهند و بفروشند که در ازای هر خرید، ۷۰٪ قیمت نرم افزار به برنامه‌نویس تعلق می‌گیرد. برنامه‌نویسان همچنین می‌توانند نسخه‌های رایگانی از نرم افزار خود را روی APP Store قرار دهند.

۱۵-۲ زبان برنامه‌نویسی iPhone :

زبان برنامه‌نویسی iPhone ، Objective-C است، که دستورات آن شبیه C بوده و فقط قابلیت‌های زبان smalltalk به آن اضافه شده است. کامپیوتری که امکان اجرای این SDK را داشته باشد باید حداقل سیستم عامل Mac osx leopard داشته باشد.

جاوا، بزودی بر روی iPhone شرکت SUN اعلام کرده است که بزودی نسخه ماشین مجازی خود را برای اجرا بر روی iPhone ارائه خواهد کرد تا برنامه‌های JAVA ME بر روی iPhone و iPad Touh نیز قابل اجرا باشند.

۱-۱۵-۲ Net. بر روی iPhone :

امکان اجرای فریم ورک موبایل Net. بر روی iPhone ممکن نیست. زیرا iPhone از پردازنده‌ی ARM استفاده می‌کند. اما با استفاده از Monotouch می‌توان نتایج مشابهی را به دست آورد. Monotouch یکی از زیرشاخه‌های پروژه‌ی MONO است که امکان کامپایل کدهای Net. را برای اجرا بر روی پردازنده‌های ARM فراهم می‌کند.

۲-۱۵-۲ Flash بر روی iPhone :

هم اکنون iPhone و مرورگر Safari امکان نمایش فایل‌های فلش را ندارند. اما کاربران می‌توانند نسخه‌ی Flash Lite را برای دیدن فایل‌های فلش روی iPhone نصب کنند. اما مشکل همچنان باقی است. FLASH در اصل یک Virtual Machine با یک کامپایلر Just In Time است که هنگام اجرای فایل‌های فلش باید کدهای خارجی‌ای را روی iPhone اجرا کند که iPhone امکان چنین چیزی را به هیچ برنامه‌ای نمی‌دهد. شرکت ادوبی در نسخه فلش CS ۵ در

حال اضافه کردن بخشی به این نرم افزار است که امکان کامپایل کردن فایل های فلش را به Application های iPhone می دهد. بنابراین تعداد بازی های iPhone به ناگهان چندین برابر خواهد شد. با این وجود همچنان امکان مشاهده ی فایل های اصلی فلش بر روی iPhone ممکن نیست

وقتی به وبسایت های مختلف که در مورد سیستم عامل Android و iOS نوشتند سری می زنیم اولین چیزی که نظر ما را جلب می کند دعوی همیشگی طرفداران این سیستم عامل هاست که حتی بعضی اوقات به بحث و مشاجره های شدید ختم می شود! اما واقعا کدام یک بهترین هستند؟ آندروید به اصطلاح opensource یا iOS که با وجود محدودیت هایش میلیون ها کاربر در سرتاسر جهان دارد؟ در ادامه تاریخچه مختصری از ورود این دو سیستم عامل به دنیای دیجیتال را بیان خواهیم کرد و به مقایسه مزایا و معایب آن ها خواهیم پرداخت.

از لحاظ قدمت و سابقه iOS بی شک آندروید جوان تر را مغلوب می کند. تجربه تیم نرم افزاری شرکت اپل و تیم جداگانه ای که این شرکت به منظور گسترش یا با اصطلاح develop این سیستم عامل استخدام کرده است، به مراتب بیشتر و غنی تر از تیم برنامه نویسان گوگل آندروید می باشد. iOS با ورود گوشی iPhone و پخش کننده موسیقی iPod شرکت اپل به همگان معرفی شد و خیلی زود طرفداران پروپا قرصی پیدا کرد. گسترش این سیستم عامل و ورود آن به iPad که اولین تبلت اپل بود باعث محبوبیت چند برابر آن شد چرا که سابق بر آن تنها راه تجربه کردن iOS از طریق iPhone یا iPod بود ولی پس از انتشار نسخه تغییر یافته آن که منحصرا برای صفحه نمایش بزرگ iPad اختصاصی شده بود مالکان iPad هم این سیستم عامل بی نظیر را تجربه کردند.

آندروید هم آغازی شبیه به iOS داشت. با ورود گوشی های آندرویدی به بازار، که از شاخص ترین آن ها می توان به سری Samsung Galaxy اشاره کرد که با نسخه Eclair ۲.۱ عرضه شد و در بین راه آپدیت هایی به Froyo و Gingerbread هم دریافت کرد، اشاره نمود. سامسونگ با دیدن موفقیت iPad به فکر تولید محصولی مشابه افتاد که بعدها به اولین تبلت آندرویدی معروف شد. Galaxy Tab که اندازه صفحه آن ۷ اینچ بود و دقیقا از همان نسخه سیستم عامل آندروید بهره می

برد که روی گوشی Galaxy نصب بود. هرچند Galaxy Tab فروش چند میلیونی داشت اما نتوانست به رقم فروش چند ده میلیونی iPad برسد. این موضوع تیم برنامه نویسی آندروید را بر آن داشت که دوباره آستین ها را بالا زده و یک نسخه کاملا اختصاصی از این سیستم عامل را برای تبلت ها تهیه کنند. پس از مدتی نسخه تبلتی سیستم عامل آندروید روی تبلت Xoom که محصول شرکت Motorola بود روانه بازار شد. نام این سیستم عامل بهبود یافته Honeycomb بود که به نسخه سوم آندروید نیز مشهور است.

این حقیقت که کدام یک از این سیستم عامل ها موفق تر است و کدام یک در جذب رضایت و اطمینان مشتریان بهتر عمل کرده بحثی داغ و هیجان انگیز است که این روزها در هر گوشه ای از وب می توان اثری از آن پیدا کرد. طرفداران سینه چاک اپل و سیستم عامل iOS با تکیه بر محبوبیت و کیفیت محصولات اپل همیشه خود را یک سرو گردن بالاتر از هواداران سیستم عامل جوان تر آندروید می دانند. طرفداران آندروید نیز آزادی عمل فوق العاده و انعطاف پذیر بودن Honeycomb را برگ برنده ای می دانند که اپل از آن بی نصیب است.

خوشبختانه الان که در حال نوشتن این مطلب هستم، از این موضوع خاطر جمع هستم که بی طرفانه قضاوت خواهم کرد و هرگز به دلیل تعصب یا مسائل دیگر رای به نفع هیچ کدام از این سیستم عامل ها نخواهم داد. تجربه هفت یا هشت ماه کارکردن با iPad و از طرفی مدت ها کار کردن با Galaxy Tab ۱۰.۱ که از نسخه ۳.۱ آندروید بهره می برد بسیار به قضاوتی درست و علمی و منصفانه کمک خواهد کرد. در ادامه به مقایسه Android Honeycomb با iOS با توجه به فاکتورهایی از جمله سادگی رابط کاربری، جذابیت، عملکرد، تعداد نرم افزارهای موجود و محبوبیت خواهیم پرداخت.

۱-۱۶-۲ Android Honeycomb

با نگاهی به تصاویر فوق به سادگی در می یابید که iOS در مقایسه با Honeycomb به مراتب ساده تر و کار کردن با آن راحت تر است. در واقع سیاست اپل همیشه بر سادگی سیستم عامل هایش استوار بوده، چیزی که در سیستم عامل Mac نیز به وضوح دیده می شود. تمامی برنامه های موجود در تبلت به صورت یک شبکه ۴×۵ به نمایش در می آیند. به تعداد دلخواه دستکاپ خواهید داشت تا هر مقدار برنامه ای که مد نظرتان است را در آن ها جای دهید. قسمت پایینی نیز که به Dock معروف است، قابلیت جای دادن ۶ اپلیکیشن مختلف را داراست. آیا چیز دیگری هم هست؟ خیر! تقریباً تمام سیستم عامل ios در همین دستکاپ ساده و آیکون هایش خلاصه می شود. شاید پیچیده ترین قسمت این سیستم عامل بخش Setting آن باشد که شامل چندین بخش جدا شده از هم است که تنظیمات ارتباطی، تنظیمات صفحه نمایش، کی بورد، تنظیمات امنیتی و ... در آن جای می گیرد.

افراد زیادی این سادگی را یکی از مزایای ios می دانند و معتقدند این ویژگی باعث شده که افراد، با هر سن و تجربه ای به آسانی قادر به کار کردن با این سیستم عامل باشند. اما در آن طرف میدان، اوضاع طور دیگری است. افراد دیگری، که تعدادشان هم کم نیست، معتقدند که این رابط کاربری یا همان interface بسیار ساده و ابتدایی طراحی شده و اصلاً خلاقانه نیست. خصوصاً کسانی که تجربه کار کردن با windows را روی کامپیوتر های شخصی شان دارند زیاد از این موضوع راضی نیستند. آن ها مطمئناً Honeycomb را به عنوان سیستم عامل محبوب خود انتخاب می کنند چون که شباهت های بیشتری به یک سیستم عامل انعطاف پذیر دارد. وجود widget ها بر روی دستکاپ که دسترسی به یک برنامه خاص (application) را بدون نیاز به اجرای کامل آن فراهم می کنند بدون شک یکی از مزایای Honeycomb است. وجود نوارهای متعددی که این اجازه را به کاربر می دهند تا تنظیمات اصلی تبلت را بدون رفتن به بخش Setting انجام دهد، مثل خاموش یا روشن

کردن WiFi یا GPS, نیز یکی دیگر از ویژگی های خوشایند Honeycomb است که آن را از iOS متمایز می کند.

۲-۱۶-۲ جذابیت Honeycomb

بیشتر کاربران و حتی کارشناسان در این بخش سلیقه ای عمل می کنند. عده ای از طراحی iOS بسیار خوششان می آید و به آن لقب "شاهکار طراحی" را می دهند. خصوصا آن که این سیستم عامل به صورت انحصاری در اختیار اپل است و هیچ شرکت دیگری حق استفاده از آن را روی تبلت هایش ندارد و خب طبیعتا هنگامی که شما برای بار اول iOS را روی صفحه نمایش شفاف و فوق العاده iPad ببینید، بدون شک نظرتان را جلب خواهد کرد. استفاده از آیکون های گرافیکی و پرهیز از جداول یا نوارهای شلوغ در iOS به عقیده برخی این سیستم عامل را جذاب و همانطور که قبلا اشاره کردم، ساده کرده است.

آندروید نیز چندان بی توجه به این امر نیست. هرچند آندروید سیستم عامل پیچیده تری به نظر می آید، اما دارای جذابیت های خاص خود نیز هست. آیکون های گرافیکی و جلوه های تصویری مثل تصویر پس زمینه متحرک یا ویدجت های متنوع و رنگارنگ قطعا به هر چه جذاب شدن این سیستم عامل کمک می کنند. نکته قابل ذکر این است که بر خلاف iOS که تنها به شما اجازه تعویض تصویر پس زمینه را می دهد، در آندروید شما آزادی عمل بیشتری دارید و می توانید با حذف یا اضافه کردن ویدجت ها و آیکون ها، ظاهر دسکتاپ خود را مطابق میل خود تغییر دهید. در واقع در Honeycomb شما هر قسمت از صفحه تبلت را می توانید به یک چیز اختصاص دهید. بخشی برای خواندن اخبار، قسمتی برای مرور کردن اینترنت، قسمتی دیگر برای دیدن تصاویر و ... نکته جالب و خوشایند این است که تمامی این کارها را می توانید همزمان انجام دهید. قابلیتی که iOS از آن بی نصیب است!

۳-۱۶-۲ عملکرد

هنگامی که با iPad کار می‌کردم واقعا از سرعت اجرای برنامه‌ها لذت می‌بردم. اپلیکیشن‌ها روان و سریع اجرا می‌شدند و به ندرت Crash یا hang می‌کردند. شاید یکی از اصلی‌ترین دلایل محبوبیت این سیستم عامل همان عملکرد روان و سریعش باشد. اما متأسفانه iOS تحت سیاست‌های بسیار محدودکننده اپل، دارای کمبودها یا اگر مثبت‌تر نگاه کنیم، استثنائاتی نیز هست. انتقال داده‌ها و فایل‌ها تنها با استفاده از یک برنامه واسطه (iTunes) انجام می‌گیرد. به جز موارد معدودی از انتقال از طریق WiFi که آن‌هم در مورد فایل‌های پرحجم، خسته‌کننده و مشکل‌ساز است، تنها راه انتقال فایل به دستگاه شما از طریق برنامه iTunes به وسیله اتصال به اینترنت پرسرعت (به شرط وجود یک اکانت VPN) می‌باشد. البته از طریق خود تبلت هم می‌توان به این فروشگاه متصل شده و برنامه‌ها را دانلود کرد.

نکته دردآوری که برای کاربران، مخصوصا کاربران ایرانی که به استفاده از برنامه‌های مجانی و معمولا کرک شده عادت دارند، بسیار ناامید کننده است، پولی بودن بیشتر برنامه‌ها و بازی‌های خوب در فروشگاه اپل است. برای ارزان‌ترین برنامه باید حداقل ۱ دلار پرداخت کنید. مگر آن‌که هوس کرک کردن یا به اصطلاح jailbreak کردن دستگاه به سرتان بزند که آن‌هم در بعضی موارد کار می‌کند و در بعضی موارد به نابدی دستگاهتان منجر می‌شود. بارها شده که کاربران iOS برای استفاده از امکان نصب بازی و برنامه از روی DVD های خود، اقدام به کرک کردن دستگاهشان نمودند اما نتیجه این بود که دستگاه دیگر قادر به بوت شدن نبوده است! طرفداران آندروید همیشه روی این ضعف iOS که شاید پاشنه آشیل آن نیز باشد دست می‌گذارند. البته ناگفته نماند که طرفداران اپل با تمامی این ضعف‌ها کنار آمده و حتی برخی آن‌ها را لازم، ضروری و بسیار مفید می‌دانند. در واقع برای استفاده کردن از iOS ذهن شما باید شبیه ذهن مسولان شرکت اپل شود که به فایل‌های Flash به چشم فایل‌هایی پر از باگ و کدهای مخرب نگاه می‌کنند. یا این که انتقال فایل به صورت آزاد را بسیار خطرناک می‌دانند و تنها از طریق یک برنامه واسطه و تحت نظارت اپل به شما اجازه انتقال داده‌ها را می‌دهند. iOS حتی امکان Bluetooth را نیز محدود کرده است و به دستگاه شما اجازه نمی‌دهد که با هر موبایل یا لپ‌تاپ یا هر وسیله دیگری که از بلوتوث استفاده

میکنند، متصل شود. شما در iOS تنها می توانید با دستگاه های ساخت شرکت اپل و تعداد محدودی وسیله دیگر مثل بلندگوی استریو از طریق بلوتوث تماس پیدا کنید. وقتی این محدودیت ها را می بینید دیگر تعجبی ندارد که این نکته را هم ذکر کنم که استفاده کردن از Flash Memory نیز کاملا مردود است مگر این که دستگاه را کرک کنید و یک کیت اتصال نیز به صورت جداگانه خرید کنید.

در این زمینه آندروید Honeycomb حرف های زیادی برای گفتن دارد. Honeycomb در واقع هیچ کدام از این محدودیت ها را ندارد و به شما اجازه می دهد که کاملا آزادانه از سیستم عامل نهایت استفاده را ببرید. هر چند این آزادی عمل زمینه را برای ورود برنامه های مخرب مثل ویروس ها و اپلیکیشن های مشکل دار به تبلت شما فراهم می کند اما بسیاری از کاربران مزایای باز بودن سیستم عامل را بسیار بیشتر از معایب آن می دانند. آندروید Honeycomb به شما اجازه اتصال از طریق بلوتوث به هر دستگاهی را می دهد. نقل و انتقال فایل ها، حتی برنامه های کرک شده بسیار آسان است و سیستم عامل هیچ واکنشی به برنامه های دزدی (نرم افزارهایی که قانون کپی رایت آن ها نقض شده است) نشان نمی دهد.

درست شبیه ویندوز ۷ یا نسخه های قبلی آن که برای کاربران ایرانی ایده آل بودند. با وجود این مزیت ها آندروید Honeycomb در عملکرد دارای ضعف هایی نیز هست. معمولا تبلت هایی که این سیستم عامل را اجرا می کنند بسیار قدرتمند هستند و از پردازنده های ۲ هسته ای با فرکانس ۱۰۰۰ تا ۱۴۰۰ گیگاهرتز استفاده می کنند. اما در بسیاری از موارد شاهد اشکالاتی جزئی اما آزار دهنده هستیم. برای مثال در Samsung Galaxy tab ۱۰.۱ با این که تبلت از سخت افزار قوی بهره می برد، اما به نظر می آید که بار پردازشی نرم افزار ها به درستی روی CPU و ROM منتقل نمی شود که این باعث به وجود آمدن Lag یا همان گیر کردن حین اجرای برخی برنامه ها می شود. در واقع عدم هماهنگی کامل بین Honeycomb و سخت افزار باعث می شود که نتوانیم از تمام ظرفیت و پتانسیل پردازنده یا حافظه رام استفاده کنیم. این موضوع در iOS عملا مشاهده نمی شود و سخت افزار و سیستم عامل بسیار هماهنگ هستند که حاصل آن عملکرد روان و سریع برنامه هاست. گوگل در این زمینه بسیار کار کرده است و آپگرید ۳.۱ و مدتی بعد از آن آپگرید ۳.۲ را نیز

برای آندروید منتشر کرد که قسمتی از مشکلات گفته شده را حل کرد. اما کاملاً واضح است که آندروید هنوز هم جای پیشرفت و تبدیل شدن به یک سیستم عامل کاملاً ایده آل را دارد.

۱۷-۲ تعداد نرم افزارهای موجود

iOS در این زمینه بسیار جلوتر از Honeycomb قرار دارد. طبیعی هم هست. مدت زیادی از به بازار آمدن iPad می گذرد و در این مدت بیش از ۶۰۰۰۰ برنامه اختصاصی iPad در فروشگاه اپل قرار داده شد. این رقم در مورد Honeycomb اصلاً راضی کننده نیست و بسیار پایین تر است. البته باید این نکته را در نظر داشت که آندروید Honeycomb بسیار جوان تر از iOS است و تا به سن بلوغ برسد مطمئناً برنامه نویسان، برنامه های بیشتری برای آن تولید خواهند کرد. در حال حاضر هم تعدادی برنامه های مفید و کاربردی در Android Market پیدا می شود که نیاز کاربران را به خوبی رفع نماید. با تمام این تفاسیر، فعلاً برگ برنده از لحاظ تعداد برنامه های موجود، در دست iOS است.

۱۸-۲ محبوبیت

شاید آخرین فاکتور که باید در نظر گرفته شود محبوبیت این ۲ سیستم عامل بین کاربران است. اپل با تکیه بر مشتریان وفادار و پروپاقرص خود میلیون ها نسخه از تبلت iPad را با سیستم عامل iOS به فروش رساند. چیزی که Honeycomb از آن بی نصیب است واحد بودن تولید کننده سیستم عامل و تولید کننده سخت افزار است. سیستم عامل آندروید تولید شرکت گوگل است که روی تبلت های زیادی وجود دارد و طبیعتاً کیفیت یا عدم کیفیت هرکدام از این تبلت ها (از نظر سخت افزاری) در محبوبیت سیستم عامل تاثیر می گذارد. اگر گوگل خود نیز توانایی ساخت سخت افزار مناسب با سیستم عامل Android را داشت، شاید وضعیتی مشابه اپل پیدا می کرد اما با وجود خرید بخش تکنولوژی شرکت Motorola توسط گوگل، این شرکت فعلاً یک تولید کننده نرم افزار باقی مانده که نرم افزارهایش را به شرکت هایی مثل Samsung یا HTC می فروشد و از قبل آن درآمد زایی می کند.

نکته ای که باعث بالا رفتن آمار فروش iOS می شود کیفیت شرکت تولید کننده سخت افزار آن هست. اپل همیشه به کیفیت ساخت و شیک بودن محصولاتش اهمیت خاصی می دهد و این موضوع باعث شده که کاربران تمایل بیشتری به خرید محصولات آن و در نتیجه سیستم عامل آن داشته باشند. هرچند با ادامه همکاری گوگل با شرکت های تولید کننده سخت افزار مثل Samsung که در تولید از رقیبان جدی اپل محسوب می شوند، آینده بسیار روشنی برای سیستم عامل نوپا و پر پتانسیل Honeycomb می توان پیش بینی کرد. سیستم عاملی که در نسخه تلفن همراه، کم کم در حال ربودن گوی سبقت از اپل و سیستم عاملش می باشد و تصور کردن چیزی مشابه برای نسخه تبلت هم دور از ذهن به نظر نمی رسد. اپل هم با این وجود دست روی دست گذاشته و به زودی آخرین آپدیت از سیستم عامل iOS که ویرایش ۵ است را منتشر می کند. مطمئنا در آینده شاهد پیشرفت و بهبود کارایی هر دو سیستم عامل خواهیم بود که با گسترش سریع تبلت ها در دنیای دیجیتال هر روز تعداد کاربرانشان بیشتر و بیشتر خواهد شد.

مقایسه ای بین سیستم عامل های iOS، اندروید و ویندوز فون: کدامیک برای شما مناسب تر

گروه خبری GSM: با نزدیک شدن به پایان سال میلادی، بازار تلفن های هوشمند مانند دیگر محصولات الکترونیکی داغ تر می شود در حال حاضر تلفن های بسیار خوبی با سیستم عامل های مختلف در بازار موجود هستند. اما اگر قصد خرید تلفن هوشمند جدید را دارید بهتر است سیستم عامل های مختلف را بشناسید و براساس سیستم عامل مورد نظر خود تلفن جدید را انتخاب کنید. در ادامه چند نکته کلیدی در مورد سیستم عامل های رایج بازار (اندروید، آی او اس و ویندوز فون) را مشاهده خواهید کرد تا راحتتر انتخاب کنید.

آیا تلفن جدید اپل، iPhone S بهتر از تلفن های اندرویدی است؟ ویندوزفون یا بلک بری چگونه؟ اینها سئوالاتی هستند که در ذهن اکثر افرادی که قصد خرید تلفن هوشمند دارند مطرح می شوند. در حال حاضر سیستم عامل اندروید و پس از آن، iOS اپل پرفروش ترین های بازار هستند اما ویندوز فون چگونه است؟

برای رسیدن به پاسخ، راهنمای مختصر و کارآمدی که در ادامه آمده است را مطالعه کنید تا با نقاط قوت سیستم عامل های iPhone, Android و Windows Phone آشنا شوید.

iPhone S بصورت مختصر

سیستم عامل این گوشی ios 5 اپل است.

با سه حجم حافظه مختلف: ۱۶ GB یا ۳۲ GB یا ۶۴ GB عرضه می شود.

به خوبی با iTunes, iCloud و محصولات دیگر اپل سازگار است.

سایز صفحه نمایش لمسی آن ۳.۵ اینچ می باشد که کمی کوچکتر از استاندارد فعلی تلفن های هوشمند (۴ اینچ) رایج بازار است.

رابط کاربری: ساده است اما نمی توان آن را خیلی شخصی سازی کرد.

قابلیت های کلیدی: دوربین ۸ مگاپیکسلی باکیفیت عالی، دوربین برای مکالمه ویدیویی در جلو، دستیار هوشمند Siri

تلفن بعدی iPhone 5، تاریخ عرضه مشخص نیست، اما انتظار می رود در تابستان سال ۲۰۱۲ باشد.

Android بصورت مختصر

سیستم عامل گوگل برای ابزارهای موبایل

گوشی های متنوع در بازه های مختلف قیمتی از گوشی های ارزان تا گوشی های گران و پرقدرت با این سیستم عامل عرضه می شوند.

تمام تلفن های اندرویدی با قابلیت های یکسان ساخته نمی شوند. بعنوان مثال برخی از آنها با دوربین بسیار قوی و صفحه نمایش با کیفیت عرضه می شوند و برخی دیگر با دوربین و صفحه نمایش معمولی و گاهی کم کیفیت تولید می گردند.

به خوبی با سرویس های گوگل از جمله Google Music, Gmail, Calendar: و غیره سازگار است.

رابط کاربری: هر کدام از تولید کنندگان تلفن های اندرویدی رابط کاربری خاص خود را دارند پس برای یادگیری هر کدام کمی زمان نیاز خواهید داشت.

تلفن بعدی اندرویدی Samsung Galaxy Nexus: که به زودی با نسخه ۴ اندروید وارد بازار خواهد شد.

نسخه بعدی Android ۴.۰: Ice Cream Sandwich که با Galaxy Nexus عرضه خواهد شد. همچنین در اوایل سال ۲۰۱۲ تلفن های جدید اندرویدی نیز با این نسخه سیستم عامل وارد بازار خواهند شد.

Windows Phone بصورت مختصر

سیستم عامل مایکروسافت برای تلفن های هوشمند

در حال حاضر اکثر تلفن های مجهز به WP ۷ در بازه میانی قیمت و با بازده قابل قبول عرضه می شوند. حداقل دوربین تلفن های ویندوز فون، ۵ مگاپیکسل است.

به خوبی با Zune, Xbox Live و دیگر سرویس های مایکروسافت مانند Microsoft Office و فضای آنلاین ذخیره سازی SkyDrive سازگار است.

رابط کاربری: ساده، سراسر و دوست داشتنی اما برخی امکانات مخفی هستند.

قابلیت های کلیدی: رابط کاربری تمیز، بارکد خوان داخلی، تشخیص دهنده موزیک داخلی، یکپارچه شدن با ایکس باکس، کنترل توسط صدا

تلفن بعدی WP ۷: نوکیا Lumia ۸۰۰ که به زودی وارد بازار جهانی می شود.

نسخه بعدی سیستم عامل: مشخص نیست در حال حاضر Mango ۷.۵ Windows Phone جدیدترین نسخه است.

۱۹-۲ سؤال و جواب های اندروید

چرا نسخه های جدید اندروید با تاخیر برای تلفن های مبتنی بر این سیستم عامل منتشر می شوند؟ و آیا جدیدترین نسخه اندروید یعنی Ice Cream Sandwich این مشکل را حل خواهد کرد؟

چون نسخه های جدید اندروید ابتدا توسط تولید کنندگان ابزارهای اندرویدی آزمایش می شوند و سپس رابط کاربری و گرافیک آنها براساس استانداردهای تولیدکنندگان طراحی و ساخته می شوند، این عملیات معمولا زمانبر است. پس تاخیر در عرضه نسخه های جدید اندروید برای هر گوشی مستقیما به تولید کننده مرتبط است. با این وجود در نسخه جدید اندروید، Ice ۴.۰ Android Cream Sandwich نیز این مشکل حل نخواهد شد. در ماه می گوگل به همراه تولید کنندگان کلیدی ابزارهای اندرویدی توافق کردند که با یکدیگر همکاری کنند تا نسخه های جدید اندروید سریعتر برای ابزارهای قدیمی منتشر گردند اما متاسفانه هیچ کدام به قول و قرار خود پایبند نبودند و هنوز مشکل تاخیر در عرضه نسخه جدید وجود دارد.

من قصد خرید یک تلفن هوشمندی اندرویدی را دارم کدامیک مناسب من است Galaxy Nexus : یا HTC Rezound یا Motorola Droid Razr ؟

پیشتر جدول مقایسه ای این سه تلفن را منتشر کردیم. من نمی توانم به این سؤال پاسخ قاطع دهم، شما می بایست براساس نیاز خود تلفن مناسب خود را انتخاب کنید. به عنوان مثال اگر بدنبال دوربین، عکس و فیلم برداری هستید می بایست Rezound و یا Razr را انتخاب کنید. اما هر سه تلفن سریع و قدرتمند می باشند و جزء بهترین های اندروید به شمار می روند.

Galaxy Nexus و Droid Razr باریک و سبک هستند اما Galaxy Nexus و Rezound نمایشگر باکیفیت تری دارند Galaxy Nexus. دوربین ۵ مگاپیکسلی دارد Droid Razr. از دو

تلفن دیگر شکیل تر است و **Rezound** با هدفون فوق العاده با کیفیت **Beats** عرضه می شود .
Galaxy Nexus نیز اولین تلفن مجهز به **Android ۴.۰** است.

۲۰-۲ سوال و جواب های آیفون

با وجود اینکه **iPhone ۴ S** وارد بازار شده است آیا بهتر نیست تا عرضه **iPhone ۵** صبر کنیم؟ من شنیده ام که **iPhone ۵** در تابستان وارد بازار خواهد شد.

اگر می توانید تحمل کنید بهتر است منتظر **iPhone ۵** بمانید چون شایعات حاکی از آن است که آیفون ۵ یک گوشی متفاوت با طراحی و قدرت فوق العاده خواهد بود از طرفی **iPhone ۴ S** تفاوت ظاهری با **iPhone ۴** ندارد پس شاید بهتر باشد تا عرضه نسخه جدید صبر کنید.

۲۱-۲ سوال و جواب های ویندوز فون

کار کردن با کدامیک ساده تر است **Windows Phone ۷**: یا **iOS ۵** یا **Android ۴.۰**؟

ویندوز فون ۷ تمیزترین سیستم عامل موبایل است. کار کردن با آن بسیار ساده است و به راحتی می توانید به امکانات گوشی دسترسی داشته باشید. استفاده از نرم افزارها و رابط کاربری **WP ۷** ساده تر از دو سیستم عامل دیگر است. رابط کاربری **Metro** در ویندوز ۸ نیز استفاده شده است و به این ترتیب با کارکردن با **WP ۷** خود را برای ویندوز ۸ نیز آماده خواهید کرد. البته چند فوت و فن مخف نیز در این سیستم عامل وجود دارد به عنوان مثال می توانید با نگه داشتن یک "عنوان زنده" آن را **Pin** کنید. تعداد صفحات اصلی در ویندوز فون تنها دو صفحه است و امکان ندارد موردی را در این دو صفحه گم کنید.

در **ios** و **Android** دستتان باز تر است و می توانید صفحات زیادی را پر از نرم افزارهای مختلف کنید. در مجموع کار کردن با ویندوز فون از آی او اس و اندروید ساده تر است و کار کردن با **iOS** نیز ساده تر از اندروید می باشد. اما امکانات شخصی سازی و سفارشی کردن اندروید بسیار بیشتر از دو سیستم عامل دیگر می باشد.

آیا تمام تلفن های Windows Phone ۷.۵ با پردازنده تک هسته ای تولید خواهند شد؟ و آیا این موضوع بر روی بازده تلفن تاثیر می گذارد؟

در حال حاضر تمام تلفن های ویندوزی تک هسته ای هستند اما این بدین معنا نیست که WP ۷ از پردازنده های دو هسته ای و یا بیشتر پشتیبانی نمی کند. ویندوز فون ۷ به نحوی امور را پردازش می کند که یک پردازنده تک هسته ای به خوبی از پس نیازهای این سیستم عامل برمی آید. ولی بدون شک با توجه به مسابقه سرعتی که تلفن های هوشمند در آن در حال رقابت هستند به زودی تلفن های چند هسته ای ویندوز فون نیز عرضه خواهند شد.

آیا تنوع و تعداد نرم افزارهای WP ۷.۵ نیز به اندازه ios یا Android خواهد رسید.

تعداد نرم افزارهای Windows Phone رشد بسیار خوبی داشته است. تعداد App های ویندوز فون در فاصله زمانی بسیار کوتاهی از ۱۸ هزار عنوان به ۴۰ هزار رسیده است. این بدین معناست که مایکروسافت توانسته نظر تولید کنندگان نرم افزار را به سیستم عامل خود جلب کند. البته این تعداد نرم افزار هنوز در مقایسه با ios با بیش از ۵۰۰ هزار App و اندروید با بیش از ۳۰۰ هزار عنوان فاصله بسیار زیادی دارد.

طول عمر باتری Android و ios و WP ۷

از آنجای که طول عمر باتری یکی از مهمترین مسائل تلفن های هوشمند است آیا هیچ کدام از این سه سیستم عامل در مدیریت مصرف انرژی با مشکل مواجه می شوند؟ کدامیک مصرف کمتری دارد؟

تقریباً هر سه سیستم عامل به میزان یکسانی باتری مصرف می کنند در واقع نحوه مدیریت انرژی توسط سیستم در طول عمر باتری تاثیر گذار است. اما نه به اندازه مشخصات سخت افزاری، تجربه نشان می دهد که پردازنده های مختلف به میزان متفاوتی انرژی مصرف می کنند. در واقع اگر بعنوان مثال یک ابزار اندرویدی بیشتر از iPhone مصرف می کنند دلیل آن مشخصات سخت افزاری آن است. معمولاً پردازنده های جدیدتر مصرف کمتری نیز دارند.

فصل سوم

راه اندازی پروژه در **Xcode**

۱-۳ راه اندازی پروژه در **xcode**

اولین پروژه ما در فولدر **02-Hello world** قرار دارد. اگر می خواهید پروژه خودتان را بسازید فولدر جدیدی برای نگهداری تمام پروژه های ثبت شده تان ایجاد کنید. تمام پروژه های جدیدتان را در آن فولدر ذخیره کنید. حالا **xcode** را در قسمت **Developer/Application/** قرار دارد، اجرا کنید. به طور حتم فولدری به نام **developer** در بالای درایو هارد، خواهید یافت. در داخل این فولدر، فولدر دیگری به نام **Application** وجود دارد که در آن برنامه ای به نام **xcode.app** قرار گرفته است.

قبل از شروع کار با **xcode**، آیکون ها را به **dock** هم **drag** کنید. شما از هر دو برنامه در این پروژه استفاده خواهید کرد. هنگامی که برای اولین بار **xcode** را اجرا می کنید با یک پنجره خوش آمدگویی مواجه می شوید. حالا می توانید ساخت یک پروژه جدید را انتخاب کنید. به برنامه آموزشی **Getting standard with xcode** و یا به وب سایت **Apple Developer connection** بروید. که مستقیماً شما را به پروژه های جدیدی که ساخته اید و لیستی که در سمت راست پنجره آن قرار دارد و پروژه ها در آن جا نمایش داده می شوند، خواهد برد. چنانچه این اولین پروژه شماست، پس آن لیست خالی خواهد بود.

چنانچه ترجیح می دهید که پنجره خوش آمدگویی در هر زمانی که **xcode** اجرا می شود، دیده نشود، تنها کافی است که چک باکس مربوط به **show this window when xcode launches** را بدون تیک کنید. اگر کنجکاوید که نتیجه کار را ببینید، بعد از اتمام کار، پنجره را ببندید.

نکته:

در صورتی که وب سایت **Apple Developer connection** بروید، برنامه **ios developer** را خواهید دید، که اگر قصد شما ساخت برنامه کاربردی و ذخیره آن در **App store** باشد، قطعاً باید به این برنامه ملحق شوید. زمانی که عضو این برنامه شدید باید برای تأیید شدن، کمی وقت صرف کنید. اگر این کار را همین الان هم انجام دهید ارزش وقت صرف کردن را دارد.

نکته:

در صورتی که ابزار **ios** دارید که به یک ماشین وصل شده، ممکن است زمانی را که برای اولین بار **xcode** را اجرا می کنید با ارسال یک پیام از شما بپرسد که آیا می خواهید از آن دستگاه برای ارتقاء استفاده کنید یا نه؟ سپس پنجره ای با عنوان **Organizer**، باز شده و لیستی که برای ابزارهای مورد نیاز شما طراحی شده، ظاهر خواهد شد. فعلاً بر روی دکمه **Ignore** کلیک کنید و یا در صورت باز شدن پنجره **Organizer** آن را ببندید. اگر شما عضو برنامه پرداختی **ios developer** شوید در آن صورت به برنامه ای که چگونگی ارتقا و تست ابزار **ios** را به شما می گوید، دسترسی خواهید داشت.

با انتخاب **Application** در زیر سرفصل **ios**، آیکون های زیادی را در پن سمت راست بالایی خواهید دید که هر کدام نشان دهنده یک **project template** جداگانه بوده و در برنامه های **ios** می توان از آن ها به عنوان یک نقطه شروع استفاده کرد. آیکونی که با جمله **View-based Application**، لیبل شده است یکی از ساده ترین و نیز اولین قالبی است که در ابتدای فصل های متعدد از آن استفاده خواهیم کرد. سایر آن ها، کدها و یا منابع اضافی را که برای ایجاد واسط های متداول در برنامه **ios** نیاز داریم، فراهم می کنند.

بر روی آیکون **View-based Application** کلیک کرده و از منوی پاپ آپ **Product** که در وسط پنجره قرار دارد، گزینه **iphon** را انتخاب کنید. با این کار به **Xcode** گفته می شود که هدف ما اجرای این برنامه در آیفون و سائز مخصوص به صفحه نمایش آن است. هم اکنون آیفون و آی پد تنها ابزارهایی هستند که به وسیله این قالب پشتیبانی می شوند. در حال حاضر، به آیفون می پردازیم.

در ادامه، بر روی دکمه **Choose ...**، کلیک کنید. در اینجا از شما درخواست می شود که پروژه جدید را با استفاده از صفحه استاندارد ذخیره کنید. به عنوان نام پروژه، **Hello World** را تایپ کرده و در هر کجا که تمایل دارید، آن را ذخیره کنید. فولدر **Document** هم برای این کار بد نیست، اما شاید بخواهید برای پروژه **Xcode** خود یک فولدر مشخص کنید.

۲-۳ پنجره پروژه Xcode

بعد از اینکه بر روی گزینه save کلیک کردید، Xcode ایجاد شده، و پروژه شما باز می شود و یک پنجره جدید از پروژه نیز باز می شود. هنگامی که پنجره پروژه برای بار اول ساخته می شود برای انجام کارها، کمی کوچک به نظر می رسد بنابراین معمولاً پنجره را گسترده تر می کنیم. اطلاعات زیادی در این پنجره گردآوری شده و اینجا جایی است که شما زمان زیادی برای ارتقاء iOS خواهید گذراند.

در بالای پنجره پروژه، نوار ابزاری قرار دارد که توسط آن به بسیاری از مواردی که معمولاً استفاده می شوند دسترسی خواهید داشت. در زیر نوار ابزار، پنجره به ۳ بخش اصلی و یا پن، تقسیم می شود. پنی که در سمت چپ و پایین پنجره قرار دارد، Groups & files نام دارد و تمام منابعی که پروژه شما را ایجاد می کنند در اینجا گروه بندی شده اند. تعدادی از تنظیمات مربوط به پروژه نیز در این قسمت قرار دارند. همان طور که در Finder، با کلیک کردن بر روی مثلث کوچکی که در سمت چپ هر آیتم قرار دارد، می توان زیر مجموعه های آن را باز کرد، با کلیک دوباره بر روی آن مثلث نیز می توان تمام آن ها را پنهان کرد. پن سمت راست در بالای صفحه، Detail view (یا فقط Detail) نام دارد که اطلاعات دقیقی درباره مجموعه های انتخاب شده در پن Groups & files را به شما نشان می دهد. پن پایینی در سمت راست Editor نام دارد. چنانچه یک فایل تکی را چه در Groups & files و چه در Detail، انتخاب کنیم Xcode می داند که این نوع فایل را چگونه باید نمایش دهد. محتویات فایل نیز در پن Editor نشان داده خواهد شد. فایل های قابل ویرایشی مانند کدهای منبع را نیز می توان در اینجا ویرایش کرد. در واقع اینجا مکانی است که می توانید کدهای منبع برنامه تان را نوشته و ویرایش کنید.

حالا از اصطلاحات فنی صحبت به میان آمد، نگاهی به پن Groups & files می اندازیم. اولین آیتم در لیست باید با نام پروژه تان یکی باشد (در اینجا یعنی Hello World). این آیتم برای کدهای منبع و سایر منابع برنامه به عنوان یک امتیاز محسوب می شود. در این موقعیت به استثنای آیتم هایی که در زیر Hello World قرار گرفته اند، به فکر سایر آیتم های موجود در قسمت Groups & files نباشید. توجه کنید که مثلث Disclosure در سمت چپ فایل Hello

World باز است و در آن چند فولدر فرعی به نام های frameworks, Resources, Other Sources, Classes و Products وجود دارد. اجازه دهید به طور مختصر درباره هر کدام از این فولدر توضیح دهیم.

- **Classes**: جایی است که شما بیشتر وقت خود را در آن جا می گذرانید. بیشتر کدهایی را که می نویسید به آن جا می رود در واقع تمام کلاس های **objective-v** به آن جا تعلق دارد. این اختیار را دارید که فولدرهای فرعی دیگری در زیر فولدر **Classes** بسازید تا به سازمان دهی کدهایتان کمک کنید. از این فولدر در فصل بعدی استفاده خواهیم کرد.
- **Other Sources**: این فولدر شامل فایل های از کدهای منبع است که جزء کلاس **objective-v** نیستند. معمولاً زمان زیادی را در این فولدر نخواهید گذارند. تنها زمانی که پروژه جدیدی در برنامه کاربردی آیفون ایجاد می کنید، ۲ فایل در این پوشه به وجود خواهد آمد.
- **Resources**: شامل فایل های **noncode** است که در برنامه وجود دارند اینجا جایی است که فایل هایی مثل آیکون های برنامه و دیگر تصاویر و نیز فایل های صوتی در آن قرار دارند.
- **Frameworks**: نوع خاصی از کتابخانه (**library**) می باشند که شامل کدها و منابعی مانند عکس ها و فایل های صوتی هستند. هر قالب (**Framework**) و یا کتابخانه ای که به این فولدر اضافه کنید به برنامه کاربردی تان متصل می شود و کدهای شامل قادر به استفاده از آبجکت ها، دستورالعمل ها و منابعی که در کتابخانه و یا قالب ها وجود دارد، هستند.
- **Products**: شامل برنامه ای است که پروژه شما، هنگامی که کامپایل می شود، ایجاد می کند. اگر فولدر **Products** را باز کنید آیتمی به نام **Hello world.app** را خواهید دید. این برنامه ای است که این پروژه خاص، آن را ایجاد کرده است. در حال حاضر **Hello world.app** به رنگ قرمز باشد، به این معنی است که فایل یافت نشده است. تا زمانی که هنوز پروژه مان را کامپایل نکرده باشیم، نمایش دادن نام یک فایل به رنگ قرمز روشی است که از طریق آن **xcode** به ما می گوید که نمی تواند فایل را بیابد.

نکته:

folders موجود در پن Groups & file الزاماً پاسخگوی سیستم فایل های Mac تان نیست. این ها دسته بندی منطقی ای در xcode هستند که به سازمان دهی داده ها کمک کرده و باعث می شود، آن چه را که در برنامه کاربردی به دنبالش هستید، سریع تر و آسان تر پیدا کنید. چنانچه به فولدرهای پروژه تان روی هارد درایو نگاهی بیاندازید، متوجه خواهید شد که وقتی وقتی دارای فولدر Classes باشیم، دیگر فولدری به نام other sources و یا Resources وجود نخواهد داشت. غالباً آیتم هایی که شامل فولدرهای این دو پروژه هستند، در دایرکتوری root پروژه ذخیره می شوند ولی شما می توانید آن ها را در هر جای دیگری حتی خارج از فولدر پروژه ذخیره کنید. سلسله مراتب وجود در xcode کاملاً به مرتبه بندی سیستم فایل، بستگی دارد.

به عنوان مثال بردن یک فایل به خارج از فولدر classes موقعیت فایل را در xcode درایو هارد شما تغییر نمی دهد.

۳-۳ معرفی interface Builder (سازنده واسط)

سازنده واسط (interface Builder) معمولاً با علامت اختصاری IB نوشته می شود.

در لیست Group \$ files مربوط به پنجره پروژه، گروه Resources را باز کنید. در صورتی که بر روی فایل Hello-world view controller.xib دابل کلیک کنید، آن فایل در IB باز خواهد شد. اگر این اولین باری است که از IB استفاده می کنید، یک پنجره دسته بندی شده، ظاهر می شود. چنان چه قبلاً از آن استفاده کرده باشید، این پنجره در جایی قرار می گیرد که آخرین بار آن را ترک کرده اید.

پنجره های که Hello-world view controller.xib نام دارد پنجره اصلی nib می باشد. اینجا home اصلی و نقطه شروع این فایل است. به جز دو آیکن اول (File's , first responder , Owner) بقیه آیکن های این پنجره، نمایانگر نمونه تکی، از کلاس Objective-C می باشد که وقتی فایل nib بارگیری می شود به صورت خودکار برایتان ایجاد می گردد. فایل nib مان دارای

یک آیکون اضافی و مود نیاز برای `File's owner` و `First Responder` می باشد. آیکون سوم نیز ارائه کننده آبجکت صفحه است.

IB دارای تاریخچه طولانی می باشد و از سال ۱۹۹۸ تاکنون، برای ارتقای برنامه های کاربردی `Mac OSK` , `Open STEP` , `Next STEP` استفاده شده و اکنون نیز برای آیکون بکار گرفته می شود. **IB** از دو نوع فایل پشتیبانی می کند؛ فایل هایی با فرمت قدیمی تر و با پسوند `(nib)`، و فرمت جدیدتر با پسوند `(xib)`، قالب های پروژه آیفون همگی به طور پیش فرض از فایل های `.xib` استفاده می کنند. اما تا همین اواخر، تمام فایل های **IB** دارای پسوند `nib` بودند. به همین دلیل اکثر برنامه نویس ها، آن فایل ها را `nib` می نامیدند. اکنون نیز فایل های **IB**، صرف نظر از اینکه پسوند آن ها `.xib` بوده و یا `.nib`. اکثراً فایل `nib` خوانده می شوند. در واقع اپل از عبارت `nib` یا فایل `nib` در تمامی سندهای خود استفاده می کند.

با نوشتن کد می توانید یک دکمه بسازید اما معمولاً از **IB** برای ساخت دکمه و مشخصاتش استفاده می شود. فایل `Hello-world view controller.xib` که در حال حاضر مورد نظر ماست، بعد از آغاز برنامه، به طور خودکار `load` می شود. فعلاً نگران چگونگی این کار نباشید. پس اینجا مکان خوبی است برای ساخت آبجکت هایی که واسط کاربر شما را درست می کنند. به عنوان مثال برای اضافه کردن یک دکمه به برنامه کاربردی تان، لازم است که آبجکتی از نوع `UI Button` معرفی کنید. می توانید این کار با نوشتن کد انجام دهید. مانند این خط:

```
UIButton *myButton = [[UIButton alloc] initWithFrame:aRect];
```

در **IB** نیز می توانید دقیقاً همین کار را با `drag` کردن یک دکمه از پالت آبجکت های `Interface` در پنجره اصلی برنامه کاربردی خودتان انجام دهید. **IB** تنظیم ویژگی های دکمه را آسان تر می کند زمانی که برنامه شما آغاز می شود و از آن جایی که دکمه در فایل `nib` ذخیره می شود، وقتی برنامه آغاز به کار کند، این دکمه به طور خودکار معرفی خواهد شد.

همان طور که در قبل گفتیم پنجره `Hello-world view controller.xib` پنجره اصلی فایل `nib` است. هر فایل `nib` با دو آیکون `File's Owner` , `First Responder` آغاز به کار کند و نیز نمی توان آن ها را پاک کرد. پس می توانید حدس بزنید که این دو آیکن بسیار مهم هستند.

`File's Owner` همیشه اولین آیکون در هر فایل `nib` بوده و نشان دهنده آبجکتی است که فایل `nib` را از دیسک `load` کرده است. به عبارت دیگر `File's Owner` آبجکتی است که مالک این کپی از فایل `nib` می باشد.

آیکون دوم در این فایل و در واقع در تمامی فایل های `nib` , `First Responder` نامیده می شود، اما `First Responder` آبجکتی است که کاربر بیشتر با آن سر و کار دارد. به عنوان مثال اگر کاربر اخیراً داده ای را به فیلد متن وارد کند، آن فیلد، `First Responder` (اولین پاسخگوی) جاری خواهد بود. اولین پاسخگو در تعاملات کاربر با واسط تغییر می کند و این آیکون راه مناسبی را برای برقراری ارتباط با هر آن چه که کنترل و یا مشاهده کنید می باشد.

سایر آیکون های موجود در این پنجره، غیر از آن دو مورد خاص، نشان دهنده یک نمونه از آبجکتی است که در زمان `load` شدن فایل `nib` ایجاد می شوند.

آیکون `view` نمونه ای از کلاس `UI view` را ارائه می کند، یک آبجکت `UI view` محدوده ای است که کاربر می تواند آن را ببیند و با آن در ارتباط باشد. این آیکن نشان دهنده هر چیزی است که کاربر می تواند در برنامه ما آن را ببیند.

نکته: در واقع از لحاظ فنی، برنامه کاربردی ما نیز بیشتر از یک `view` دارد. تمامی اجزاء واسط کاربر که قابل نمایش بر روی صفحه باشند، شامل دکمه ها، فیلدهای متن، و لیبل ها همگی زیر کلاس های `UI view` هستند.

۴-۳ اضافه کردن یک لیبل به صفحه

سمت راست آیکون اینجا مکانی است که می توانید تمام داده های موجود در `Cocoa Touch` را در آن بیابید و `IB` نیز از آن پشتیبانی می کند. `Darg` کردن یک آیتم از کتابخانه به پنجره فایل `nib`.

یک نمونه از کلاس را به برنامه شما اضافه خواهد کرد. چنان چه پنجره کتابخانه را ببندید، می توانید با انتخاب گزینه `Tool > library` آن را مجدداً ظاهر نمایید. گزینه های این پالت همگی از `IOS UI kit` می باشند که آن هم قالبی است از آبجکت هایی که برای ایجاد واسط کاربر در برنامه، از آن ها استفاده می شود.

`UI kit` همان نقشی را در `Cocoa Touch` دارد که `Apple kit` در `Cocoa` ایفا می کند. ولی به دلیل تفاوت در برنامه هایشان حقیقتاً می توان تفاوت های زیادی در بین آن ها مشاهده کرد. به عبارت دیگر اصول کلاس هایی از قبیل `NSString` و `NSArray` در میان `Cocoa` و `Cocoa Touch` یکسان می باشد.

یک لیبل نشان دهنده متن کوچکی است که بر روی صفحه یک ابزار `ios` نمایش داده می شود، اما نمی توان آن را مستقیماً توسط کاربر ویرایش کرد.

به دلیل این که آبجکت های واسط کاربر، سلسله مراتبی هستند ما لیبل مان را به عنوان یک زیرصفحه به صفحه اصلی اضافه می کنیم (صفحه ای به نام `View`)

`IB` بسیار هوشمندانه عمل می کند. چنان چه یک آبجکت، زیر صفحه ها را نپذیرد، قادر نخواهید بود که سایر آبجکت ها را به آن `drag` کنید. `Drag` کردن یک لیبل از کتابخانه به داخل صفحه ای به نام `View`، باعث می شود که یک نمونه از `UI label` به عنوان زیر صفحه به صفحه اصلی برنامه کاربردی مان اضافه گردد.

یک لیبل را از کتابخانه `drag` کرده و به پنجره `View` ببرید.

بیاید لیبل را ویرایش کنیم که حرف های زیادی برای گفتن دارد. بر روی لیبل که ساخته اید دابل کلیک کرده و متن `Hello world` را تایپ کنید. سپس لیبل را به هر جایی که می خواهید روی صفحه قرار بگذارید، `drag` کنید. با ذخیره کردن آن کارتان تمام شده است. `File > save` کرده و سپس به `xcode` برگردید و برنامه تان را ساخته و اجرا کنید.

در Xcode گزینه **Build > Build & Run** را انتخاب کنید. با این کار **xcode** برنامه ما را کامپایل کرده و در سیمولاتور آیفون اجرا می کند.

وقتی کارهای عملی را انجام دادید، حتماً از سیمولاتور خارج شوید. **Xcode**، **IB** و سیمولاتور هر کدام یک برنامه کاربردی مجزا هستند.

۱-۴-۳ توصیه

چنانچه ابزار ios تان به **Mac** متصل باشد، ممکن است که برنامه ها در هنگام ساخت و اجرا، آنطور که برنامه ریزی شده اند اجرا نشوند. به طور خلاصه برای اینکه بتوانید برنامه هایتان را روی آیفون بسازید و اجرا کنید، باید **sign up** کرده و یکی از برنامه های **Apple IOS developer** را خریداری نمایید و سپس از طریق فرآیند پیکربندی کردن **xcode**، آن را به طور دقیق و مناسب اجرا کنید. هنگامی که عضو این برنامه بشوید، اپل اطلاعاتی را که برای انجام این کار نیاز دارید، ارسال خواهد کرد. در ضمن بیشتر برنامه های این پروژه به خوبی در سیمولاتور اجرا می شوند. اگر ابزارتان به برق وصل است، قبل از اینکه **Build & Run** را بزنید این گزینه را انتخاب کنید:

Project > set Active SDK > simulator

۵-۳ تغییر مشخصه ها

به **IB** برگشته و بر روی لیبل **Hello world** تک کلیک کنند، طوری که انتخاب شده باشد. حالا با انتخاب **Inspector > Tool**، پنجره ای به نام **Inspector** باز می شود. جایی که می توانید مشخصات آبجکت انتخاب شده جدید را تنظیم کنید. با کمک **Inspector** می توانید چیزهایی مثل اندازه، رنگ، سایه و بسیاری موارد دیگر را تغییر دهید. **Inspector** کارهای ظریفی انجام می دهد مثلاً اگر شما یک فیلد متن انتخاب کنید، **Inspector** ویژگی های قابل ویرایشی را برای آن فیلد به شما نشان خواهد داد. اگر یک دکمه را انتخاب کنید ویژگی های قابل ویرایش دکمه را خواهید دید، و غیره.

به کارتان ادامه دهید و شکل ظاهری لیبل را هر طوری که می پسندید تغییر دهید. سپس آن را ذخیره کرده و به `x code` بر گردید و **Build & Run** را مجدداً انتخاب کنید. تغییراتی را که انجام داده اید، هم اکنون باید در برنامه کاربردی تان نمایان شوند. دیدید؟ یک بار دیگر بدون نوشتن هیچ کدی تمام این کارها را انجام دادیم. **IB** با کمک طراحی گرافیکی واسط، شما را از صرف کردن وقت اضافی برای نوشتن کدهای خاص، خلاص می کند. به دلیل اینکه برای نوشتن کدهای طولانی و خسته کننده به منظور ساخت واسط کاربر، باید وقت صرف کنید.

اکثر محیط های پیشرفته برنامه های مدرن، ابزاری دارند که به کاربر اجازه ساخت واسط های گرافیکی را می دهد. فرق بین **IB** و ابزارهای دیگر در این است که **IB**، کدی را تولید نمی کند که مجبور به نگهداری اش باشد. در عوض آبجکت هایی را براساس **Objective-C** ایجاد می کند. درست مانند همان کاری که شما در کدگذاری انجام می دهید. و سپس آبجکت ها را در فایل **nib** به این منظور که بتوانند در زمان اجرا، مستقیماً در حافظه **load** شوند، سریال بندی می کنیم. این کار بسیاری از اشکالاتی را که در هنگام کدگذاری پیش می آید متوقف می سازد، و به طور کلی یک روش قدرتمند است.

۶-۳ جلوه دادن به آیفون- پرداخت حرکات لمسی

قبل از اتمام این فصل بیایید کمی آب و رنگ به برنامه کاربردیمان بدهیم تا کمی احساس کنیم که مثل یک برنامه **IOS** درست و حسابی است. در ابتدا پروژتان را اجرا کنید. هنگامی که پنجره سیمولاتور ظاهر شد، بر روی دکمه **home** در سیمولاتور آیفون کلیک کنید (منظور دکمه مشکی با یک مربع سفید که در انتهای ترین قسمت پنجره قرار دارد)، با این کار به صفحه اصلی آیفون بر می گردید.

به آیکون **Hello world** در بالای صفحه نگاه کنید، آیا این آیکون هرگز عمل می کند؟ به منظور درست کردن آن احتیاج به ساخت یک آیکون و ذخیره کردن آن به عنوان یک فایل (**.png**) که مخفف (**portable network graphic**) می باشد، دارید. اندازه آن باید در حدود **۵۷×۵۷** پیکسل باشد. لازم نیست که سعی کنید شکل دکمه ها را دقیقاً همان طور که قبلاً در آیفون بوده، شکل دهید.

چون آیفون شما به طور خودکار لبه هایش را تکمیل کرده و به آن ظاهر شیشه ای ماندی می دهد.
یک سطح مربع شکل معمولی را بسازید.

نکته:

برای آیکون برنامه کاربردی تان، باید از یک تصویر با فرمت (png) استفاده کنید. در واقع باید از این فرمت برای تمام تصاویری که به پروژه IOS خودتان اضافه می کنید، استفاده کنید. اگر چه اکثر عکس ها با سایر فرمت های معمولی به خوبی نشان داده می شوند. اما شما حتماً باید از فرمت (png) استفاده کنید مگر اینکه دلیل قانع کننده ای برای استفاده از فرمت های دیگر داشته باشید. X code به طور خودکار، تصاویر (png) را در زمان ساخت، برای اینکه سریع تر و کارآمدتر باشند، بهینه سازی می کند.

بعد از اینکه آیکون برنامه کاربردی تان را طراحی کردید، فایل png را از Finder به فولدر Resources در X code، drag کنید.

و یا می توانید فولدر Resources را در X code انتخاب کرده و ... add to project > project را برگزینید و آن را به فایل تصویری آیکونتان هدایت کنید. وقتی این کار را کردید X code چند مورد را به شما نشان خواهد داد. می توانید کپی کردن مستقیم فایل توسط X code، به داخل پروژه را انتخاب کرده و یا فقط آن را به عنوان یک مرجع به فایل اصلی اضافه کنید. ولی به طور کلی ایده خوبی است که منابع را در پروژه X code خودتان کپی کنید. مگر اینکه فایل مورد نظر در پروژه های دیگر نیز مشترک باشد.

هرگاه انواع معمولی فایل را به پروژه تان اضافه می کنید. X code می داند که با آن چه کاری باید انجام داد. در نتیجه آن فایل تصویری را به صورت خودکار در برنامه شما کامپایل می کند، بدون اینکه لازم باشد کار اضافی انجام بدهید.

کاری که تا کنون انجام داده ایم، ثبت تصویر `icon.png` در داخل پروژه بوده که نتیجه اش ورود یک تصویر ساخته شده به برنامه باندل می باشد. قدم بعدی، تعیین این تصویر خاص به عنوان آیکون برنامه است.

در پن `Group & Files` مربوط به پنجره پروژه `x code` فولدر `Resources` را باز کرده و بر روی فایل `Hello-world-info.plist` کلیک کنید. این فایل لیست پروپرتی است و شامل اطلاعات کلی موجود در برنامه می باشد.

هنگامی که آیکون `Hello-world-info.plist` را انتخاب می کنید، لیست پروپرتی در پن ویرایش ظاهر خواهد شد. در داخل لیست پروپرتی، ردیفی را با لیبل فایل `icon`، که در ستون سمت چپ قرار دارد، بیابید. در ستون سمت راست، سلول مطابق با آن ردیف، باید خالی باشد. با دابل کلیک بر روی سلول خالی، به نام فایل که هم اکنون به پروژه اضافه کردید، `png` را اضافه کنید.

نکته:

اگر ورودی `icon file` را در `plist` نادیده بگیرید، باز هم آیکون شما نمایش داده می شود. به طور پیش فرض، چنان چه هیچ نامی برای فایل آیکون در نظر گرفته نشده باشد، `SDK` در یک منبع `icon.png` به جستجو پرداخته و از آن استفاده می کند.

۳-۷ آمادگی جهت کامپایل کردن و اجرای برنامه:

قبل از اینکه برنامه را کامپایل و اجرا کنیم، نگاهی به ردیف های دیگر `Hello-world-info.plist` می اندازیم تا ببینیم آیا این تنظیمات به خوبی که باید باشند هستند یا نه؟ لازم است که به یکی از آن ها به نام `Bundle identifier` توجه بیشتری داشته باشیم. این یک شناسه بی نظیر برای برنامه کاربردی است که باید همیشه در برنامه نصب شود.

اگر می خواهید برنامه کاربردی تان را روی سیمولاتور `IOS` اجرا کنید نام قراردادی استاندارد `Bundle identifier` را باید از یکی از بالاترین حوزه اینترنتی از قبیل `com` و یا `org` استفاده کنید. سپس اسم شرکت یا سازمانی را که شما پشتیبانی می کند و بعد نام برنامه کاربردی تان را قرار دهید.

اگر می خواهید برنامه کاربردی تان را در یک ابزار IOS واقعی اجرا کنید، ایجاد شناسه باندل برنامه شما، نیاز به انجام فرآیند بیشتری دارد. چنان چه عضویت در برنامه IOS SDK را خریداری کرده باشید، می توانید درباره موضوع یاد شده در [iphone Program Portal](#) بخوانید. حالا که اینجاییم، از فرصت استفاده کرده و بر روی کلمه [Yourcompany](#) در شناسه باندل موجود، دابل کلیک می کنیم و آن را به [apress](#) تغییر می دهیم. مقادیر شما در انتهای ردیف، یک کد خاص است که در زمان اجرای برنامه با نام برنامه کاربردی تان جایگزین می شود. و این امکان را به شما می دهد که شناسه باندل برنامه را با نامش هماهنگ کنید.

زمانی که این تغییر انجام شد، برنامه را کامپایل و اجرا کنید. هنگامی که کار اجرای سیمولاتور به اتمام رسید دکمه مربع سفید را فشار دهید تا به صفحه [home](#) بروید و آیکون جالب خود را ببینید. شاید متوجه شده باشید که برنامه [Holle world](#) متفاوت را نشان می دهد. یکی با آیکون کلی و دیگری با آیکون جدید و درخشان. این به چه معناست؟

شناسه باندلی را که هم اکنون تغییر داده اید، چیزی است که منحصراً شناسه یک برنامه کاربردی خاص است. هنگامی که [Your company](#) را به [apress](#) تغییر می دهید، در واقع به IOS می گوئید که این یک علامت جدید و یک برنامه کاربردی متفاوت است.

نکته:

وقتی که می خواهید برنامه کاربردی قدیمی را از صفحه [Home](#) مربوط به سیمولاتور آیفون، پاک کنید، می توانید به آسانی فولدري به نام [iphone simulator](#) را از فولدر [Application support](#)، که در فولدر [library](#) دایرکتوری [home](#) قرار دارد، پاک کنید. همچنین می توانید سیمولاتور را با انتخاب گزینه زیر، [reset](#) کنید.

Ios simulator > Reset content & setting ...

۳-۸ آوردن آن به صفحه **home**

کمی به عقب برگردید. اگر چه ممکن است کارهایی را که در این فصل انجام دادیم. ناچیز به نظر برسد، اما در واقع اصول بسیاری را پوشش داده ایم. در این فصل درباره قالب پروژه های IOS و ایجاد یک برنامه کاربردی یاد گرفتید و دیدید که چگونه از **Interface Builder** استفاده می شود و اینکه چگونه می توان آیکون برنامه و شناسه باندل را تنظیم کرد.

با وجود این، **Hello-world** یک برنامه یک جانبه محض است: ما به کاربر اطلاعاتی را ارائه می دهیم. اما هرگز از آن ها هیچ ورودی، دریافت نخواهیم کرد. هر زمان آمادگی پیدا کردید که ببینید چگونه ورودی ها را از کاربر ابزار ios گرفته و چطور اعمالی را بر مبنای آن ورودی ها انجام می دهیم.

فصل چهارم

سرگرمی بیشتر با واسط کاربر

۱-۴ سرگرمی بیشتر با واسط کاربر

در این فصل برنامه ای را خواهیم ساخت که اطلاعات کاملتری درباره کنترل ها به شما می دهد.

یک نمای تصویر (image view)، یک اسلایدر، و دو متن متفاوت برای لیبِل، یک کنترل سگمنت و تعدادی کلید و یک دکمه ios که بیشتر شبیه به یک دکمه ios واقعی است، اجرا خواهیم کرد. یاد می گیرید که چگونه از صفحه سلسله مراتبی برای گروه بندی آیتم های چندتایی تحت یک صفحه منشاء معمولی، استفاده کنید و چگونه دستکاری واسط را در زمان اجرا آسان تر کنید. خواهید دید که چگونه مقادیر کنترل های مختلف را با استفاده از خروجی ها و آرگومان sender مربوط به متدهای اکشن، تنظیم کرده و بازخوانی کنید. سپس نگاهی می اندازیم به سطوح اکشن تا کاربر را وادار به انتخاب کرده و هشدار برای اهمیت بازخورد به کاربر ارائه دهیم. درباره حالت های کنترل و استفاده از تصویرهای بسط یافتنی برای ساخت دکمه ها نیز آگاهی پیدا خواهید کرد.

به دلیل این که در این فصل از آیتم های مختلفی از واسط کاربر استفاده می شود، قصد داریم کمی متفاوت تر از فصل قبلی کار کنیم. برنامه کاربردی را به چند قسمت تقسیم می کنیم. یک قسمت را در یک زمان اجرا کرده و دوباره به عقب برمی گردیم و در بین سیمولاتور ios و x code، فورت (forth) می کنیم، قبل از حرکت به قسمت بعدی، هر قسمت را امتحان می کنیم. با فرایند تقسیم بندی در ساخت یک واسط پیچیده و تبدیل آن به قسمت های کوچک تر، ترسناک بودن آن را از بین می بریم. چرخه کدنویسی، کامپایل، اشکال زدایی، قسمت اعظم کار یک برنامه نویس امروزی خواهد بود.

۲-۴ صفحه نمایشی پر از کنترل

همان طور که گفتیم برنامه ای که قصد داریم در این فصل بسازیم کمی پیچیده تر از آن است که در فصل قبل ساختیم. ولی هنوز از یک صفحه و کنترلر واحد، استفاده خواهیم کرد.

لوگویی که در بالای صفحه آیفون قرار دارد، یک image view می باشد که در این برنامه کاربردی بیشتر از نمایش یک تصویر ثابت ندارد. در زیر لوگو، دو فیلد متن وجود دارد: یک امکان وارد کردن

متن النبای عددی و دیگری فقط امکان وارد کردن اعداد را می دهد. در زیر فیلد متن، اسلایدر قرار دارد زمانی که کاربر اسلایدر را حرکت دهد، مقادیر لیبل نزدیک به آن تغییر می کند به نحوی که همیشه مقادیر اسلایدر منعکس نشان داده شوند. در زیر اسلایدر، یک کنترل سگمنت شده، و دو سوئیچ قرار دارد. کنترل سگمنت شده، بین دو نوع مختلف از کنترل ها تغییر وضعیت می دهد که کلید تغییر وضعیت در زیر آن قرار گرفته است. زمانی که برنامه کاربردی برای بار اول اجرا می شود، دو سوئیچ در زیر آن ظاهر می شود که تغییر مقدار هر سوئیچ مقدار سوئیچ دیگری را به نسبت آن تغییر می دهد. احتمالاً این آن کاری نیست که می خواستید در یک برنامه کاربردی واقعی انجام دهید. اما به شما نشان داده می شود که چگونه مقادیر کنترل را به صورت برنامه ریزی شده تغییر دهید و چگونه Cocoa Touch، بدون این که کاری انجام بدهید، اکشن های خاص را فعال می کند.

وقتی کاربر کنترل سگمنت شده را می زند چه اتفاقی می افتد. سوئیچ ها ناپدید شده و یک دکمه جایگزین آن ها می شود. وقتی که دکمه **Do something** زده شود یک صفحه اکشن ظاهر شده و از کاربر می پرسد که آیا واقعاً تمایل به زدن آن دکمه دارد یا نه؟. این یک روش استاندارد برای نشان دادن واکنش به ورودی می باشد که ممکن است خطری داشته باشد. و به کاربر فرصتی داده می شود که جلوی اتفاقات بد را بگیرد. اگر گزینه **Yes, I'm sure** را انتخاب کنید برنامه کاربردی اطلاع می دهد و به کاربر امکان این را می دهد که بداند همه چیز **ok** است.

۳-۴ کنترل های فعال و غیر فعال

کنترل های واسط کاربر، در سه شکل اساسی قرار گرفته اند: **Active** و **passive** دکمه هایی که در فصل قبلی به کار گرفتیم بهترین مثال از کنترل های فعال می باشند. زمانی که آن دکمه ها را می زدید یک اتفاق می افتاد، معمولاً قسمتی از کدها شروع به کار می کردند.

بعضی از کنترل ها می توانند در یک روش غیر فعال کار کنند و به سادگی در مقادیری که کاربر وارد می کند، تا زمانی که آماده فعالیت شوند نگهداری می شوند. این کنترل ها متدهای اکشن را راه نمی اندازند ولی کاربر از طریق مقداردهی به آن ها می تواند با این کنترل ها کار کند. بهترین مثال برای کنترل غیر فعال، یک فیلد متن، در صفحه وب می باشد. اگر چه وقتی که به بیرون از فیلد ضربه

بزنید، کدهای ارزیابی می توانند شروع به کار کنند، اکثریت فیلدهای متن در صفحه وب، با کلیک بر روی دکمه submit، به سادگی اطلاعات را به server می دهند. در حقیقت فیلدهای متن خودشان هیچ کدی را راه اندازی نمی کنند اما با کلیک کردن بر روی دکمه submit، داده های فیلد متن، این کار را انجام می دهند.

در یک ابزار ios بیشتر کنترل های قابل دسترس، می توانند در هر دو روش استفاده شوند. اما بستگی به نیاز شما دارد. تمام کنترل های ios زیر مجموعه UI control می باشند و به همین دلیل توانایی راه اندازی متدهای اکشن را دارند و بیشتر کنترل ها نیز می توانند به صورت غیر فعال استفاده شوند. اما بعضی از کنترل ها مانند دکمه ها در این روش به طور حتم مقاصد زیادی را تأمین نمی کنند مگر این که در یک روش فعال استفاده شوند تا این که کدها را راه اندازی کنند.

ممکن است متوجه شده باشید که چند تفاوت بین عملکرد کنترل های ios و کنترل های ابزار Mac وجود دارد. در اینجا چند مثال می زنیم:

- به دلیل وجود واسط multiTouch، تمام کنترل های ios می توانند چندین متد اکشن را بسته به این که چگونه لمس شوند راه اندازی کنند. ممکن است یک اکشن متفاوت تنها با یک ضربه انگشت کاربر، برای تمام کنترل ها راه اندازی شود.
- می توانید با زدن یک دکمه، یک اکشن را فعال کرده و با برداشتن انگشت از روی دکمه، اکشن دیگری را فعال کنید.
- می توانید تنها از یک کنترل برای فرمان دادن به چندین متد اکشن در یک رویداد استفاده کنید. همچنین می توان دو متد اکشن مختلف را برای رویداد touch up inside فعال کرد، یعنی هر دو متد زمانی که انگشت کاربر بعد از لمس دکمه برداشته می شود، فراخوانی می شوند.

تفاوت بزرگ دیگری که بین ios و Mac وجود دارد این است که به طور معمول ابزارهای ios صفحه کلید فیزیکی ندارند (مگر این که خود شما یک صفحه کلید external به آن وصل کنید) صفحه کلید استاندارد ios در واقع یک صفحه می باشد که با یک سری از دکمه های کنترل پر شده

است. کدهای شما احتمالاً هرگز به طور مستقیم با این صفحه کلید iOS تعامل ندارد. اما همان طور که کمی بعد در این فصل خواهید دید، گاهی اوقات نیاز است کدهایی را برای صفحه کلید بنویسید تا به روشی که شما می خواهید عمل کند.

۴-۴ ساخت برنامه کاربردی

اگر X code را قبلاً باز نکرده اید آن را اجرا کرده و یک پروژه جدید به نام Control Fun بسازید. قصد داریم مجدداً از الگوی View-basedApplication استفاده کنیم. پس پروژه را همان گونه که در فصل قبلی انجام دادید، بسازید.

اکنون که پروژه تان را ایجاد کردید، بیایید یک تصویر از آن در image view استفاده خواهیم کرد را به دست آوریم. قبل از اینکه تصویر در IB برای کاربر قابل دسترس باشد، باید آن را به x code گزارش داد. پس هم اکنون آن را گزارش می کنیم. می توانید از یک تصویر با فرمت png. در آرشیو پروژه در فولدر 04 Control Fun استفاده کنید و یا از هر تصویری که خودتان می خواهید استفاده کنید. چنان چه از تصویر خودتان استفاده می کنید، از اینکه فرمت آن png. باشد اطمینان پیدا کنید. اندازه آن نیز باید برای فضای موجود مناسب بوده و باید کمتر از ۱۰۰ پیکسل، طول و حداکثر ۳۰۰ پیکسل، عرض داشته باشد که به راحتی و بدون نیاز به سایزبندی داشته باشد در قسمت بالای صفحه قرار بگیرد.

تصویر را به فولدر Resources پروژه تان اضافه کنید همان طور که در فصل قبل با drag کردن تصویر از Finder به فولدر Resources انجام دادیم.

اجرای نمای تصویر (image view) و فیلدهای متن

با اضافه کردن تصویر به پروژه، بدم بعدی را که اجرای پنج جزء واسط در بالای صفحه نمایش برنامه می باشد، بر می داریم.

نمای تصویر، دو فیلد متن و دو لیبل.

۵-۴ تعیین خروجی ها

قبل از این که شروع به ساخت این GUI کنیم، باید مشخص کرد که کدام یک از آبجکت ها نیاز به خروجی دارند. به خاطر داشته باشید که خروجی ها قبل از این که به هر چیزی در ویرایشگر nib ارتباط پیدا کنند، باید در فایل سرفصل کلاس کنترلر، تعریف شوند.

این نمای تصویر فقط یک تصویر ثابت است. قصد داریم آن را طوری مشخص کنیم که مستقیماً در IB نمایش داده شود و همچنین در زمان اجرای برنامه، تصویر هیچ تغییری نکند. به این ترتیب نیازی به خروجی ندارد. در صورتی که بخواهیم تصویر و یا یکی مشخصه ها در زمان اجرا تغییر کند باید یک خروجی برای آن مشخص کنیم. اما در این جا نیازی به این کار نداریم. این مورد درباره لیل موجود نیز صدق می کند آن ها برای نمایش متن در آن جا قرار داده شده اند ولی در زمان اجرا تغییر نمی کنند و کاربر هیچ ارتباطی با آن ها نخواهد داشت پس برای آن ها نیز احتیاجی به خروجی نخواهیم داشت.

از طرف دیگر چنان چه نتوانیم به داده هایی که در فیلدهای متن قرار دارند، دسترسی پیدا کنیم در واقع زیاد از این دو فیلد استفاده ای نخواهد شد. راه دسترسی به داده های نگهداری شده توسط یک کنترلر غیر فعال، استفاده از یک خروجی می باشد. پس لازم است که یک خروجی برای هر کدام از این فیلدهای متن تعیین کنیم. انجام این کار در حال حاضر کهنه و قدیمی محسوب می شود. به این ترتیب چرا دو خروجی به همراه پروپرتی های مربوط به آن ها را با استفاده از نام های numberfield , namefield به فایل کلاس control-FunviewController.h اضافه نکنیم. بعد از اتمام کار، فایل control-FunviewController.h باید چیزی شبیه به این باشد.

```
#import <UIKit/UIKit.h>

@interface Control_FunViewController : UIViewController {
    UITextField * nameField;
    UITextField * numberField;
}

@property (nonatomic, retain) IBOutlet UITextField *nameField;
@property (nonatomic, retain) IBOutlet UITextField *numberField;

@end
```

قبل از آن که به فایل nib پردازیم بیایید دستورالعمل های @synthesize را به فایل control-FunviewController.m اضافه کنیم:

```
#import "Control_FunViewController.h"
@implementation Control_FunViewController
@synthesize nameField:
@synthesize numberField:
...
```

نکته:

علامت (...) را در انتهای کدهای لیست شده می بینید؟ از این علامت استفاده می کنیم تا به شما اطلاع دهیم که کدهای دیگری نیز در پشت آن چه که به شما نشان داده ایم، وجود دارند که به هیچ تغییری نیاز ندارند. در این فصل ما تمام کدها را به فایل اجرایی مان اضافه کرده و سپس با استفاده از این علامت، می توانیم نیاز به نشان دادن تمام فایل را بعد از اضافه کردن یک یا دو خط از کدها، کاهش دهیم.

همچنین باید مطمئن باشیم که در مورد حافظه احتیاط لازم را کرده ایم. از آن جایی که پروپرتی های namefield , numberfield را با کلمه کلیدی retain اعلام کرده ایم، لازم است که هر دوی آن ها را در متد deallocتان، release کرده و به انتهای فایل بروید و این دو خط را به متد dealloc موجود اضافه کنید:

```
- (void)dealloc {
    [nameField release];
    [numberField release];
    [super dealloc];
}
```

۶-۴ تعیین اکشن ها

به پنج آبجکتی که فصل قبل نگاه کنیم. آیا نیاز است که هیچ اکشنی را معرفی کنید؟ تا وقتی که کاربر اجازه کار با لیبل ها و صفحه تصویر را نداشته باشد، دلیلی به ساخت اکشن برای آن ها وجود ندارد. درباره آن دو فیلد متن چطور؟ فیلدهای متن، کنترل های غیر فعال کلاسیک هستند. آن ها مدت زمان زیادی مقادیر را حفظ می کنند تا شما بتوانید از آن ها استفاده کنید. ما بر روی این فایل ها هیچ ارزیابی انجام نمی دهیم غیر از این که، ورودی عددی فیلد را با نشان دادن تنها یک لایه عددی به جای صفحه کلید کلی (که می توانیم این کار به طور کامل در IB نیز انجام دهیم) محدود خواهیم کرد. پس در این موارد نیازی به اکشن نداریم. درست است؟ خوب این مطالب را تا همین جا داشته باشید و بیاید قسمت اول واسط کاربر را امتحان کنیم.

۷-۴ ساخت واسط (Interface)

مطمئن شوید که هر دو فایل را ذخیره کرده اید. فولدر Resources در پن Group & files را باز کرده و بر روی control-FunviewController.xib کلیک کنید تا IB اجرا شود. چنان چه پنجره View باز نیست، بر روی آیکن View در فایل nib پنجره اصلی دابل کلیک کنید. اکنون توجهتان را به کتابخانه جلب کنید. در صورتی که باز نیست، می توانید library را از منوی Tool انتخاب کنید و ۱/۴ از راه را در طول لیست library پایین بیاید تا اینکه image view را پیدا کنید.

۸-۴ اضافه کردن نمای تصویر (image view)

یک نمای تصویر را به داخل پنجره ای که view نام دارد drag کنید. چون این اولین آیتمی است که در view قرار داده اید، IB سعی می کند به طور خودکار آن را سایز بندی کند تا این که هم سایز صفحه شود. اگر نخواهیم که نمای تصویر کل فضا را اشغال کند، می توانیم از drag کردن برای سایز بندی نمای تصویر استفاده کنیم تا حدوداً هم سایز تصویری که در xcode وارد کرده ایم

بشود. فعلاً راجع به به دست آوردن دقیق آن سایز نگران نباشید انجام این کار در همان لحظه آسان تر خواهد بود.

به هر حال گاهی اوقات یک آبجکت انتخاب نمی شود و انتخاب دوباره آن ممکن است خیلی سخت باشد زیرا پشت آبجکت دیگری که کل صفحه را اشغال کرده، قرار گرفته است و یا هیچ حاشیه ممتدی ندارد. در این موقعیت نا امید نشوید برای انتخاب مجدد آبجکت در پنجره اصلی nib راهی وجود دارد. در پنجره اصلی، سه دکمه که با کلمه **View Mode** لیبل شده، وجود دارد. بر روی دکمه وسطی کلیک کنید تا صفحه سلسه مراتبی مربوط به nib را دریافت کنید. با انجام این کار به شما امکان رفتن به زیر صفحه ها داده می شود. دابل کلیک کردن بر روی هر کدام از آیتم های این صفحه باعث می شود که همان آیتم در پنجره اصلی انتخاب شود.

وقتی که نمای تصویر انتخاب شد، **inspector** را با فشار دکمه $\times 1$ بیرون بیاورید که در آن آپشن های قابل ویرایشی از کلاس **Ullmage view** را خواهید دید. مهمترین تنظیمات برای نمای تصویر در بالاترین قسمت **inspector** به نام **Image** قرار گرفته اند. اگر بر روی آن فلش کوچک که در سمت راست فیلد قرار دارد کلیک کنید، یک منو با تصویرهای قابل استفاده باز خواهد شد که در آن جا تمام تصویرهایی را که لازم در پروژه **xcode**تان اضافه کنید، خواهید دید. تصویری را که چند لحظه پیش اضافه کردید انتخاب کنید. این تصویر باید هم اکنون در نمای تصویرتان نمایان شود.

۱-۸-۴ سایز بندی کردن نمای تصویر

اکنون سایز نمای تصویر دقیقاً به اندازه تصویرتان است. درباره آن صحبت خواهیم کرد. یک راه آسان برای سایز بندی کردن صفحه برای هم اندازه شدن با تصویر انتخاب شده، انتخاب گزینه **size ti fit** از منوی **Layout** می باشد که به طور خودکار هر صفحه را به اندازه دقیق و مورد نیاز ما سایز بندی می کند. همچنین باید سایز بندی را تا جایی که تصویر در مرکز و بالا قرار بگیرد. به طوری که با خط راهنماهای **Horizontal Center in Container** از منوی فرعی **Alignment** در **Layout**، به راحتی یک آیتم را در مرکز صفحه قرار دهید.

نکته:

darg کردن سایز بندی صفحه در IB می تواند کار سختی باشد. دکمه **viwe mode** سلسله مراتبی را در پنجره اصلی **nid** فرموش نکنید. چون به شما کمک خواهد کرد. که نمای صفحه را پیدا کرده و انتخاب (دابل کلیک) کنید. و وقتی زمان سایز بندی کردن فرا رسید **option** را نگاه دارید. IB چند خط قرمز ن تاثیر گذار بر بروی صفحه رسم می کند که به آسان تر شدن این کار کمک میکند تا سایز دقیق نمای صفحه را دریافت کند. این رثش را **drag** کردن موثر نیست ولی اگر گزینه **Show Bounds Rectangles** را از منوی **Layout** انتخاب کنید خطی در دور آیتم های **interface** شما رسم می کند که باعث می شود راحت تر را ببینی. می توانید با انتخاب گزینه **Show Bounds Rectangles** آن خط ها را پاک کنید.

۹-۴ ویژگی **Mode**

آپشن بعدی در پایین **inspecoter** نمای صفحه ، یک منوی پاپ آپ در **Mode** می باشد. منوی **mode** تعیین می کند که تصویر چگونه در داخل صفحه تنظیم شود و آیا آن به طور مناسب در صفحه قرار گرفته و میزان شده است. در استفاده از آپشن های مختلف در منوی **mode** آزاد هستید، اما بهترین انتخاب گزینه ای **center** است. به یاد داشته باشید که انتخاب هر آپشنی که در مقیاس تصویر تاثیر بگذارد، به طور بالقوه فرایند بالاسری را اضافه خواهد کرد. پس بهتر است از آنها دوری کرده و تصویرتان را قبل از ارائه کردن به درستی سایز بندی کنید. اگر می خواهید همان تصویر را در چند سیز مختلف نشان دهید، به جای اینکه ابزار **ios** را در زمان اجرای برنامه وادار به مقیاس گذاری کنید. معمولاً بهتر است که چند کپی از تصویر را در سایز بندی های مختلف تهیه کنید.

۱۰-۴ اسلایدر **Alpha**

آیتم بعدی در **Alpha, inspector** می باشد که باید با احتیاط با آن ها کار کنید. **Alpha** تعیین می کند که چگونه تصویر را شفاف کنید. این کار باعث می شود هر مقداری کمتر از ۱.۰ باشید، آیفون شما این تصویر را طوری ناپیدا و شفاف می کند که هر شیئی که در زیر آن قرار دارد نمایان

می شود. با تعیین مقداری کمتر از ۱.۰، حتی اگر در واقع هیچ چیزی در زیر تصویر قرار نداشته باشد، باعث خواهید شد که برنامه شما پردازشگر را صرف چرخه محاسبات شفاف سازی، نماید.

پس آن را بر روی هیچ مقدار دیگری غیر از ۱.۰ تنظیم نکنید مگر این که دلیل خوبی برای انجام این کار داشته باشید.

۱-۱۰-۴ نادیده گرفتن پس زمینه

می توانید آیتم بعدی را که **Background** نامیده می شود، نادیده بگیرید. این یک پروپرتی است که از **UIView** گرفته شده است اما تأثیری بر روی ظاهر نمای تصویر نمی گذارد.

۱-۱۱ ویژگی **tag**

آیتم بعدی **tag** نام دارد که اگر چه ما در این فصل از آن استفاده نمی کنیم ولی ارزش اشاره کردن را دارد. تمام زیرمجموعه های **UIView** که صفحه ها و کنترل ها هستند، دارای یک پروپرتی به نام **tag** هستند که فقط یک مقدار عددی است و می تواند با نمای تصویر شما همراه شود. **Tag** برای استفاده شما طراحی شده است و سیستم هرگز مقادیر آن را تغییر نداده و یا تنظیم نمی کند. اگر مقادیر **tag** را به کنترل و یا یک صفحه اختصاص دهید، باید مطمئن شوید که همیشه همان مقادیر را خواهند داشت، مگر این که خودتان آن را تغییر داده باشید.

Tag یک راه ساده و مستقل از زبان است که می توان از آن بریا شناسایی آبجکت ها در **interface** استفاده کرد. به طور مثال چنان چه شما دارای پنج دکمه متفاوت باشید که هر کدام یک لیبل جداگانه دارند و بخواهید تنها از یک متد اکشن برای اداره کردن این پنج دکمه استفاده کنید، در این مورد نیاز به راهی دارید که زمانی که متد اکشن فراخوانی می شود، به تفاوت میان دکمه ها پی برده شود. مطمئناً می توانید با توجه به عنوان دکمه ها این کار را بکنید، اما هنگامی که برنامه ما به زبان دیگری مانند سانسکریت و یا **Swahili** ترجمه شود کدهای شما کار نخواهند کرد. برخلاف لیبل ها، **tag**ها هرگز تغییر نخواهند کرد. پس اگر مقادیر **tag** را در این جا تنظیم کنید می توانید در **IB**

از آن به عنوان سریع ترین و قابل اعتمادترین راه برای بررسی این که کدام کنترل به متد اکشن در آرگومان sender ارسال شده استفاده کنید.

۱۲-۴ چک باکس های طراحی

مجموعه ای از چک باکس های drawing در زیر tag قرار گرفته است. اولی چک باکسی است که با کلمه opaque لیبل شده، مطمئن شوید که تیک خورده باشد. این کار به ios اطلاع می دهد که هیچ چیز در پشت صفحه شما نباید ترسیم شده باشد و به متدهای طراحی ios امکان بهینه سازی برای سرعت بخشیدن به طراحی را می دهد. ممکن است تعجب کنید که اصلاً چه نیازی به تیک زدن چک باکس opaque وجود دارد؟ و چرا قبلاً مقدار ۱.۰ را به Alpha داده ایم تا در حقیقت، هیچ چیز!! را ناپیدا و شفاف کند؟ به این دلیل که مقادیر Alpha برای قسمت هایی از تصویر که طراحی شده در نظر گرفته شده است. اما در صورتی که یک تصویر به طور کامل یک نمای تصویر را پر نکند و یا در آن جا به دلیل وجود مجرا در Alpha و یا مسیر برش خورده، فرورفتگی هایی وجود داشته باشد، صرف نظر از مقدار تعیین شده در Alpha موضوعات در زیر تصویر نشان داده خواهند شد. با انتخاب opaque به ios اطلاع می دهیم که هیچ چیزی در زیر این تصویر نیازی به ترسیم شدن ندارد، پس نیازی به تلف کردن وقت برای پردازش هر چیزی در زیر آبجکت ها نیست. می توانیم با خیال راحت opaque را انتخاب کنیم زیرا قبلاً گزینه size to fit را زده ایم که باعث می شود نمای تصویر با سایز عکس مطابق شود. چک باکس hidden دقیقاً کاری را انجام می دهد که از نامش پیداست. چنان چه آن را تیک بزیند، کاربر نمی تواند این کنترل را ببیند. بعضی اوقات پنهان شدن کنترل می تواند مفید باشد. بعداً در این فصل به آن اشاره خواهد شد.

زمانی می رسد که سوئیچ ها و دکمه ها را پنهان می کنیم، اما مدت زیادی ان را بدون تیک نگه می دارید و یا می توانید آن را با مقادیر پیش فرض رها کنید.

چک باکس بعدی clear Context Before Drawing نام دارد که به ندرت نیاز به تیک زدن دارد. زمانی که تیک دار شود، ios کل فضای پوشیده شده توسط کنترل را با یک سیاه شفاف، ترسیم

می کند. قبل از آن که در واقع خود کنترل را ترسیم کند. و به دلیل این که به ندرت نیاز به استفاده از آن است دوباره خاموش می شود. مطمئن شوید که این چک باکس، بدون تیک باشد.

Clip subviews یک آپشن جالب است. در صورتی که صفحه شما دارای زیر صفحه باشد و آن زیر صفحه به طور کامل محدوده ها و مرزهای صفحه والدش را نداشته باشد، این چک باکس مشخص می کند که چگونه زیر صفحه ها ترسیم خواهند شد. چنان چه این چک باکس دارای تیک باشد فقط قسمت هایی از زیرصفحه ها که در محدوده های والد قرار دارند، ترسیم می شود و در صورتی که بدون تیک باشد، زیرصفحه ها به طور کامل ترسیم می شوند حتی اگر در بیرون مرزهای والد قرار گرفته باشند.

ممکن است به نظر برسد که رفتار پیش فرض باید مخالف آن چه که واقعاً هست، باشد و **Clip subviews** در صورت نیاز استفاده کنید اما این چک باکس به طور پیش فرض غیر فعال می شود. چک باکس پایانی در این قسمت **Auto resize subviews** نام دارد که به **ios** اطلاع می دهد که در صورت تغییر سایز در صفحه، زیر صفحه ها را سایزبندی کند. پس این چک باکس را تیک دار کنید. زمانی که ما به صفحه امکان تغییر سایز ندهیم، انجام این تنظیمات مهم نیست.

۱۳-۴ چک باکس های تعاملی

دو چک باکس آخر باید با تعاملات کاربر انجام شوند. اولین چک باکس، **user interaction Enabled** می باشد که مشخص می کند آیا کاربر می تواند بر روی این آبجکت ها کاری انجام دهد؟ برای بیشتر کنترل ها این باکس باید تیک دار باشد. زیرا در صورت غیر فعال بودن، کنترل هرگز قادر به راه اندازی متدهای اکشن نمی باشد. با این وجود لیبل ها و نمای تصویر به طور پیش فرض بدون تیک هستند. زیرا آن ها اغلب فقط برای نشان دادن اطلاعات ثابتی استفاده می شوند. تمام آن چه که قرار است در این جا انجام دهیم، نمایش یک عکس بر روی صفحه نمایش است و نیازی به تیک دار کردن این قسمت نیست.

آخرین چک باکس، **Multiple touch** است که مشخص می کند آیا این کنترل توانایی دریافت رویدادهای **Multiple touch** را دارد یا نه. رویدادهای **Multiple touch** امکان استفاده از حرکات پیچیده ای مانند نیشگون (**pinch**) را می دهند که برای **zoom** کردن در بیشتر برنامه های کاربردی **ios** استفاده می شوند. درباره این حرکت و رویدادهای **Multiple touch** در فصل بعد صحبت خواهیم کرد. از آن جا که این نمای تصویر، تعاملات با کاربر را نمی پذیرد، دیگر دلیلی برای استفاده کردن از **Multiple touch events** وجود ندارد پس آن را بدون تیک باقی بگذارید.

۱۴-۴ اضافه کردن فیلدهای متن

زمانی که کار با نمای تصویر به اتمام رسید، یک فیلد متن از کتابخانه برداشته و آن را به پنجره **drag, view** کنید. آن را در زیر نمای تصویر قرار داده و از خط راهنمای آبی رنگ برای تنظیم کردن تصویر یا حاشیه سمت راستی استفاده کنید، هنگامی که فیلد متن ظاهر می شود. آن خط راهنما به شما می گوید که چقدر به یک آبجکت نزدیک هستند. می توانید فعلاً فیلد متن را در همین جا رها کنید ولی به آن یک ظاهر متعادل بدهید. فیلد متن را کمی بیشتر به سمت پایین حرکت دهید. به خاطر داشته باشید که شما همیشه می توانید به **IB** برگشته و موقعیت و سایز اجزاء واسط را تغییر دهید بدون این که کدی تغییر کند و یا نیازی به برقراری دوباره کانکشن ها باشد.

بعد از این که فیلد را رها کردید یک لیبل را از کتابخانه بردارید و آن را تا جایی که با حاشیه سمت چپ صفحه هم تراز شود، بکشید و به طور عمودی نیز با فیلد متنی که قبلاً در آن جا قرار داده اید هماهنگ شود. توجه داشته باشید که وقتی لیبل را به اطراف حرکت می دهید، چند خط راهنمای آبی ظاهر می شود. این خطوط هماهنگ سازی لیبل را با فیلد متن که در بالا، وسط و پایین خط مبنا قرار دارد آسان تر می کند قصد داریم لیبل و فیلد متن را با استفاده از راهنمای خط مبنا تنظیم کنیم که خطی از انتهای نوشته لیبل به سمت فیلد متن رسم می کند، اگر خط راهنمای آبی از وسط متن لیبل رسم شود، شما در وسط خط راهنما هستید نه در راهنمای خط مبنا. استفاده از راهنمای خط مبنا باعث می شود که نوشته لیبل و نوشته ای که کاربر فیلد متن تایپ می کند در یک وضعیت عمودی مشابه در روی صفحه نمایش، قرار بگیرد.

نتیجه گیری

iOS سیستم عاملی است که با پای بندی به اصول و سیاق همیشگی اپل سعی دارد کاربر را با یک دیدگاه شفاف و مشخص آشنا کند. استفاده از این سیستم عامل ساده است و کاربر با یک چشم به هم زدن با کارکردن با آن آشنا میشود. حجم زیاد برنامه های موجود در این پلتفرم آن را از رقبای جدا می سازد و این موضوع درباره محتوای ویدئویی و صوتی نیز صدق میکند.

بنابراین اگر به دنبال سیستم عامل و تبلتی برای کارهای روزمره مانند وبگردی و بازی و خرید و گوش دادن به موسیقی و خرید یا اجاره و تماشای فیلم و یادداشت برداری و خواندن کتاب هستید . iOS و آی پد گزینه مناسبی برای شما خواهد بود . حتی برنامه های موجود این سیستم عامل و تبلت را در کارهای اداری و تجاری و بسیاری کارهای تخصصی نیز کارآمد می سازد . بنابراین اگر تبلت خود را بخصوص برای کارهای اداری و تجاری لازم دارید iOS باز هم می تواند گزینه مناسبی باشد از طرفی اگر جزء دسته خوره ها (یا به قول معروف Geek *s) هستید و دوست دارید که همه جای سیستم خود را دست کاری کنید و برنامه ها را به روش خود نصب کنید لازم است بدانید که در iOS باید در محدوده های تعیین شده حرکت کنید و چارچوب ها مشخص هستند.

برای سفارشی سازی سیستم عامل امکانات زیادی وجود ندارد و نصب برنامه یا حتی انتقال موسیقی و فیلم باید با روش از پیش تعیین شده اپل انجام شوند. در عین حال مزایای سرویس های موسیقی نماهنگ و فیلم آی تونز به عنوان محتوایی که بخش عمده ای از سرگرمی دیجیتال کاربران را تشکیل میدهد در کنار ویژگی هایی مانند i Message و مهم تر از همه این ها سرویس خدمات ابری مکملی به نام iCloud دل کندن از دنیای ios را سخت تر خواهد کرد.

مراجع:

[۱]. مهندس بهزاد رادمنش، گیتا سلیمانی - خودآموز جامع برنامه نویسی آیفون (IOS)، انتشارات

جاودانه، جنگل، چاپ اول سال ۱۳۹۱

[۲]. دانشنامه ویکی پدیا

[۳]. ماهنامه شبکه - اردیبهشت ۱۳۹۲ شماره ۱۴۴

[۴]. نشریه تایم - آوریل ۲۰۱۳ « نویسنده: هری مک کراکن » ترجمه: علی حسینی

[۵]. fa.wikipedia.org/wiki/

[۶]. www.zoomit.ir/ios

[۷]. www.zoomit.ir/ios

[۸]. www.yekmobile.com/ios.php

[۹]. appnevis.blogfa.com/post/

[۱۰]. www.asandownload.com/mobile/category/ios/

[۱۱]. www.mums.ac.ir/.../mobile_operating_systems.ppt

[۱۲]. mobileawards.ir/Default.aspx?tabid=935

[۱۳]. www.sarzamindownload.com/contents/

[۱۴]. Modern Operation Systems (Andrew S. Tanenbaum)