

Introduction to Mobile App Development

MODULE 1: Introduction to App Development

THOMPSON RIVERS UNIVERSITY | COMPUTING SCIENCE

Module 1

1. Mobile phones and network technologies
2. Different ways to develop Apps
3. Introduction to Corona

THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Mobile phones and networks technologies

THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Mobile phones and network technologies

Different types of mobile devices



Smartphones



Tablets



Android TV boxes



Smart watches

Upper left: *Ultimate Android experience* by Google,
<https://www.google.com.sg/nexus/>
Lower left: *Ultimate Android experience* by Google,
<https://www.google.com.sg/nexus/>

THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE
Upper right: *Ultimate Android experience* by Google,
<https://www.google.com.sg/nexus/>
Lower right: *Android wear* by Android,
https://www.android.com/intl/en_in/wear/

Mobile phones and network technologies

Capabilities of mobile devices



Operating System

Camera

Display

Connectivity

Sensors

Input methods



Image: Galaxy S6 edge by Samsung, <http://www.samsung.com/uk/consumer/mobile-devices/smartphones/galaxy-s/SM-G925F7KEBTU>
THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Mobile phones and network technologies

Limitations of Mobile Devices



CPU Speed

Screen size

Memory

Network Coverage

Battery

Voice recognition



Image: Galaxy S6 edge by Samsung, <http://www.samsung.com/uk/consumer/mobile-devices/smartphones/galaxy-s/SM-G925F7KEBTU>
THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Mobile phones and network technologies

Low power wireless technologies: Wi-Fi

- Wirelessly connects electronic devices that use IEEE standard for Wi-Fi 802.11x (x: a, b, g, n and ac)
- Access point of 20-70m (in doors) and 100-250m (out doors)
- Allows direct communication from one computer to another without access point intermediary (ad hoc Wi-Fi)
- Supports speeds up to 150Mbps (n) and 7Gbps (ac)



THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Mobile phones and network technologies

Low power wireless technologies: Bluetooth

- Supports both point-to-point and point-to-multipoint connections
- Piconet - 8 devices connected in Bluetooth network (1 master and 7 slaves)
- Bluetooth 3.0+HS - supports theoretical data transfer speeds up to 24 Mbit/s
- Bluetooth link - used for negotiation and establishment



THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Mobile phones and network technologies

Low power wireless technologies: Near Field Communication (NFC)

- Allows for simplified transactions, data exchange, and wireless connections between two devices in close proximity
- 2006 Nokia 6131 - first NFC phone
- Main application - mobile payment
- Support added to Android starting from 2.3 platform



THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Mobile phones and network technologies

Activity 1

Compare Wi-Fi to Bluetooth and NFC in terms of access range (distance)?

THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Mobile phones and network technologies

- **WiFi:**

An access point compliant with either 802.11b or 802.11g, using the stock antenna might have a range of 100 m (330 ft). The same radio with an external semi parabolic antenna (15 dB gain) might have a range over 20 miles.

- **Bluetooth:**

Officially Class 3 radios have a range of up to 1 metre (3 ft), Class 2, most commonly found in mobile devices, 10 metres (33 ft), and Class 1, primarily for industrial use cases, 100 metres (300 ft). Bluetooth Marketing qualifies that Class 1 range is in most cases 20–30 metres (66–98 ft), and Class 2 range 5–10 metres (16–33 ft).

- **NFC:**

NFC is a set of short-range wireless technologies, typically requiring a separation of 10 cm or less.



Different ways to develop apps

Different ways to develop apps

Native approach

- Uses native development platform and programming languages from the OS creator/maintainer to build the app.
- Android
 - Native app development platform - Android studio
 - Programming language - Java.
- iOS
 - Native app development platform - Xcode
 - Programming languages - Objective C and Swift.



THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Different ways to develop apps

Cross-platform approach

- Uses one development platform and programming language to produce different versions of the same app that targets different operating systems.
- Cross-platform tools:
 - Cordova <https://cordova.apache.org/>
 - Appcelerator <http://www.appcelerator.com/>
 - Coronalabs <https://coronalabs.com/>



THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Different ways to develop apps

Hybrid approach

- Combination of web technologies (i.e. HTML, CSS, and JavaScript).
- Hosted inside a native application that utilizes a mobile platform's WebView.
- Enables access to device capabilities (accelerometer, camera, contacts, etc.).
- Can include native UI elements where necessary.



Reference: <http://developer.telerik.com/featured/what-is-a-hybrid-mobile-app/>
Image Reference: www.liquidcanyon.com THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Different ways to develop apps

Activity 2

When should a developer choose a Native app development approach?

A developer should choose a Native approach if the complexity of the App specs requires best performance that includes fast and fluid animations as well as full access to phone hardware, multi touch support and the latest APIs.

Introduction to Corona

THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Corona

What is Corona?

- Corona is a **cross-platform** framework used to rapidly create apps and games for mobile devices and desktop systems.
- That means you can create your project once and publish it to multiple types of devices, including Apple iPhone and iPad, Android phones and tablets, Amazon Fire, Mac desktop, Windows Desktop, and even connected TVs such as Apple TV, Fire TV, and Android TV.



<https://coronalabs.com/>

THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Corona

Corona Features

- Corona is a complete framework which lets you create games, educational apps, business/utility apps, and more.
- Corona is cross-platform — develop for mobile, desktop, and connected TV devices with just one code base.
- Development is done in Lua, a powerful and easy-to-learn scripting language. Learn Lua on YouTube.
- Corona plugins give developers access to additional functionality, tools, and services.

<https://coronalabs.com/>

THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Corona

The Corona Suite

Corona Labs offers three variants of Corona so that you can use the toolset which is best for your needs:

- **Corona SDK**
 - *Corona SDK is a full-featured, cross-platform framework that enables developers to create apps rapidly and publish to all major platforms.*
- **Corona Enterprise**
 - *Corona Enterprise allows you to take advantage of Corona's incredibly fast development and still call native libraries or APIs when necessary. Corona Enterprise is ideal for bridging Corona's speed and ease-of-use with code/functionality you're already accustomed to.*
- **CoronaCards**
 - *CoronaCards is used to implement Corona inside native apps or other frameworks like Unity. This allows developers to embed Corona resources without interfering with the main application stack.*

<https://coronalabs.com/>

THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Corona

System Requirements

Corona SDK for macOS supports building applications for iOS, Android, tvOS, and macOS desktop. It requires the following basic system elements:

- macOS 10.11 or later
- Xcode (minimum version allowed by Apple to submit apps to the App Store)

Corona SDK for Windows supports building applications for Android and Win32 desktop. It requires the following basic system elements:

- *Windows 10, Windows 8, or Windows 7*
- *1 GHz processor (recommended)*
- *1 GB of RAM (recommended)*
- *OpenGL 2.1 or higher (available in most modern Windows systems)*

<https://coronalabs.com/>

THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Corona

Download SDK

- You need to provide a valid email to download the SDK

<https://coronalabs.com/>

- *You need to create an account to sign in*

<https://developer.coronalabs.com/user/login>

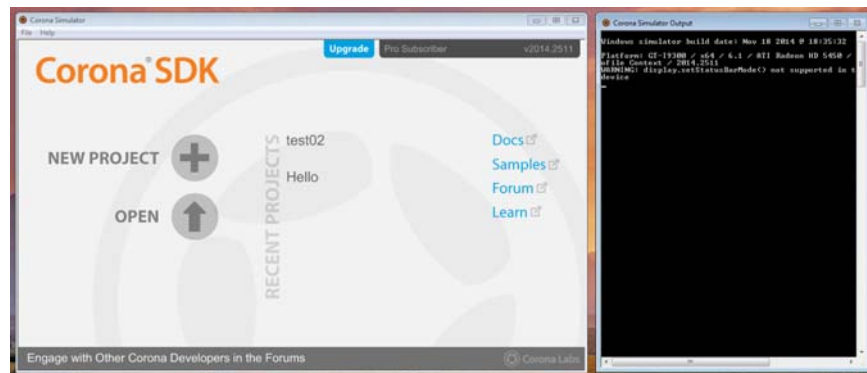
<https://coronalabs.com/>

THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Corona

Programming in Corona: Hello World

The first time you launch the Corona Terminal or Simulator it will ask you to login with your registration email that you used to create an account on the Corona Labs website. After that, create a new project

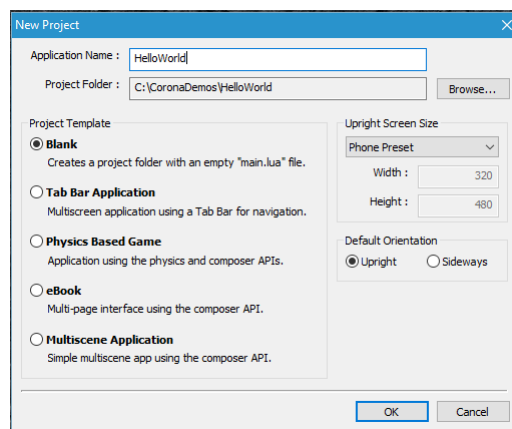


THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Corona

Programming in Corona: Hello World

Enter the Application Name: HelloWorld and Choose Blank Project Template

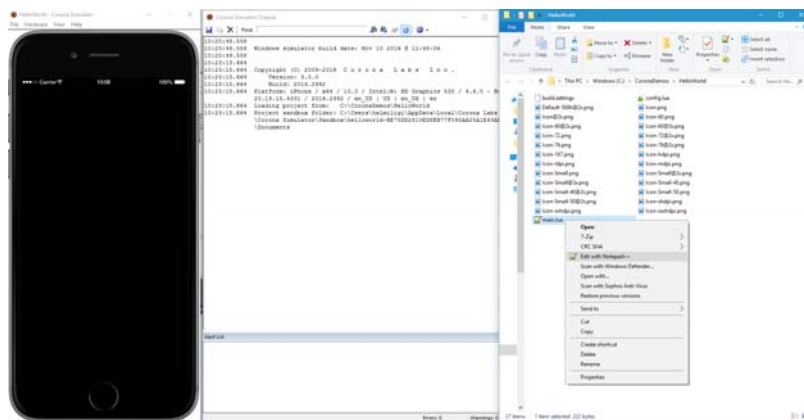


THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Corona

Programming in Corona: Hello World

You should now see three windows as shown below. Open “**main.lua**” file using your preferred editor



THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

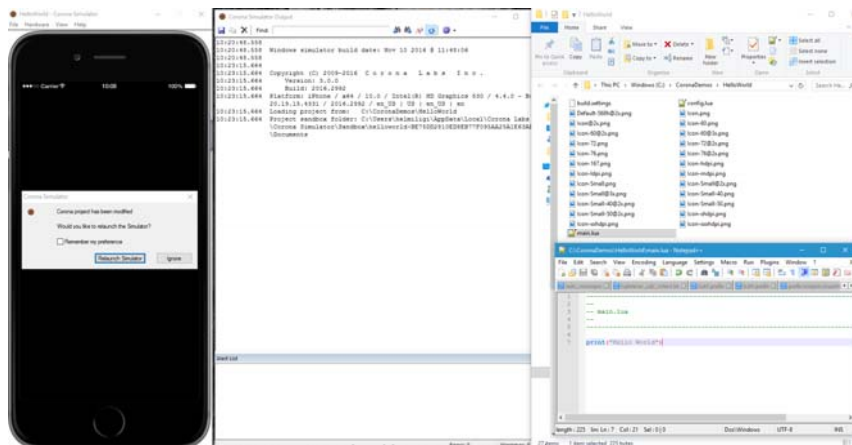
Corona

Programming in Corona: Hello World

Type:

`print("Hello World")`

And save your **main.lua** file. You should see a dialogue window. Click “**Relaunch Simulator**”

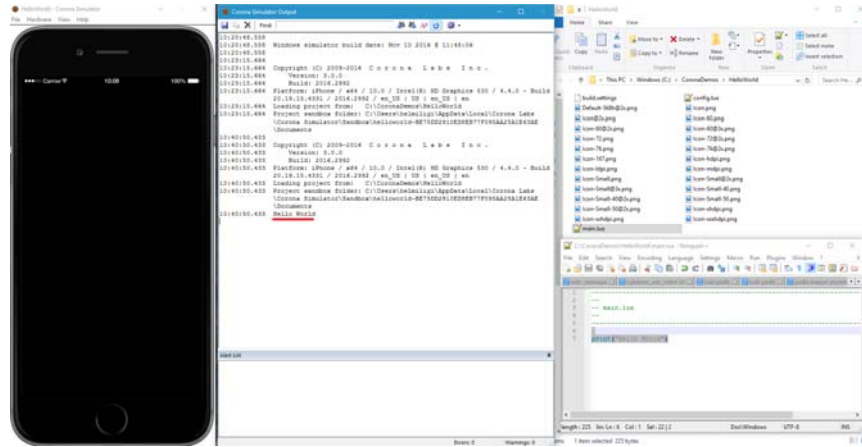


THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Corona

Programming in Corona: Hello World

You should see **"Hello World"** printed out in the terminal window.



THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

Corona

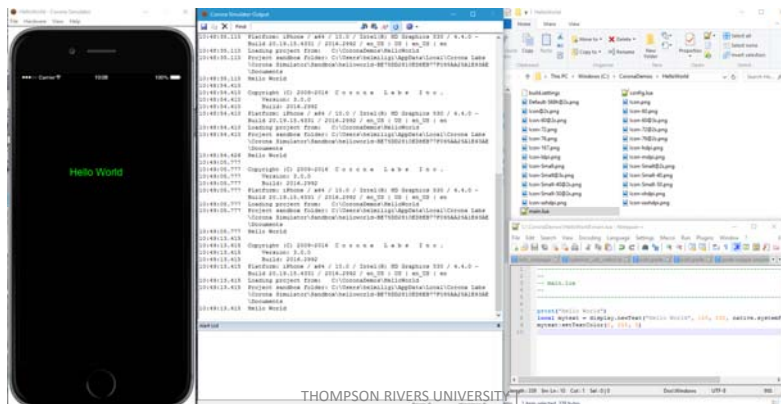
Programming in Corona: Hello World

Back in your editor (use the same file)

Type:

```
local mytext = display.newText("Hello World", 150, 150, native.systemFont, 20)
mytext:setTextColor(0, 255, 0)
```

Save the file and relaunch the simulator. **Congratulations! You have completed your first App.**



THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE

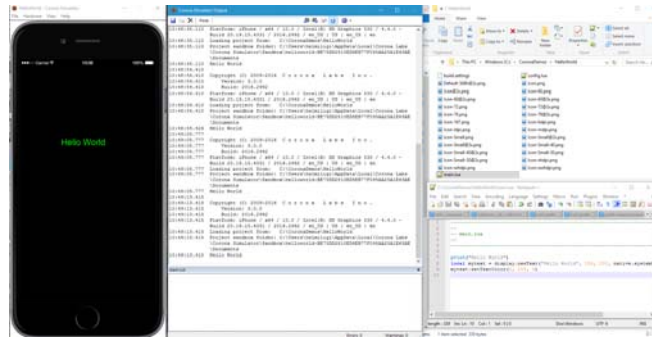
Introduction to Corona



Activity 3

Back in your editor (use the same file)

Try to change Colors, fonts, text, and location.



THOMPSON RIVERS UNIVERSITY |
COMPUTING SCIENCE



End of Module 1

THOMPSON RIVERS UNIVERSITY | COMPUTING SCIENCE