

AN INTRODUCTION TO CODING

Crescent College
Comprehensive
March 2017

Presented by
Brian Keating and Brendan Ryan

WHO ARE WE?



Brian Keating

- From Cahir, Co. Tipperary
- Studied Electrical Engineering in UCC
- Worked as software engineer for various multi-national companies, currently Intel in Shannon



Brendan Ryan

- From Kilfeacle, Co. Tipperary
- Studied Electronic Engineering in UL
- Software engineer for over 25 years including 8 years with Fujitsu in Japan, currently working for Intel in Shannon

WHY ARE WE HERE?

We believe in the importance of STEM (Science, Technology, Engineering, Mathematics)

Technical Literacy is ...

- ... a vital life skill in the 21st century
- ... good for your future job prospects
- ... good for the technology industry in Ireland
- ... good for the Irish economy and society

Software development (i.e. coding) is one key component of STEM

KEY MESSAGES

Software is everywhere!

It's easy to get started

WHY CODING?

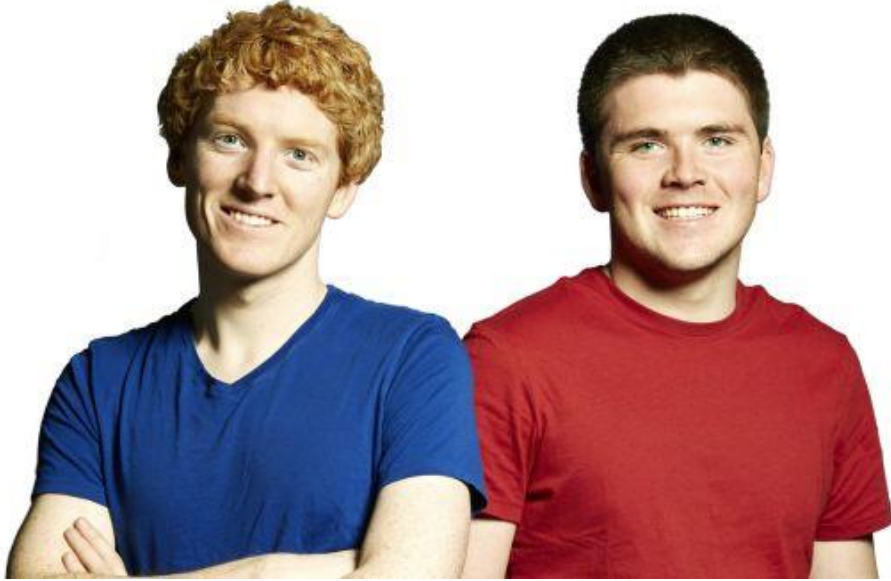
“Everybody in this country should
learn how to program a computer...
because it teaches you how to think.”
- Steve Jobs

Hear from

- Bill Gates (created Microsoft)
- Jack Dorsey (created Twitter)
- Mark Zuckerberg (created Facebook)
- will.i.am (created Black EyedPeas)
- ... and many others

“Computer Coding – What most schools don’t teach”, The Attorney Depot™,
<https://www.youtube.com/watch?v=nPblG6ceqOs>

ANYONE RECOGNISE THESE GUYS?



Patrick & John Collison

Brothers, from Limerick

Attended Castletroy College

2005: Patrick won the BT Young Scientist and Technology Exhibition, aged 16

2007: Setup Shuppa (pronounced “siopa”), selling software to optimize sales on auction sites like eBay

2008: Sold Shuppa (now Auctomatic) for €3 million

2009: John completed Leaving Cert

2010: Setup Stripe, an online payments company

2016: Stripe valued at €9 billion. Each brother worth €1 billion

The next Irish tech billionaires could be in this room!

Image taken from an article in The Irish Times in June 2016,

<http://www.irishtimes.com/business/technology/brian-caulfield-takes-issue-with-john-and-patrick-collison-1.2691357>

SOFTWARE RUNS ON ALL THESE DEVICES

Images and videos removed, showing examples of “client” devices

- Desktop computers, laptops
- Tablets and smartphones
- Smart watch
- Smart TV
- Gaming console
- Set top box
- CanSat device (satellite in a can)
- “Smart home” devices (heating systems, bulbs, security cameras)
- Electronic point of sale devices (retail industry)
- Automated Teller Machine (ATM, banking/service industry)
- Robotic arm (manufacturing industry)
- Autonomous driving (auto industry)

... AND IN THE CLOUD!

- Data centres
 - Google
 - Facebook
 - many more ...
- Networking devices
 - home router
 - core network router
 - mobile phone tower

HARDWARE VS SOFTWARE

COMPUTER HARDWARE

Processor (or CPU)

- Programmable – this is where the software runs!
- Speed measures in GHz

Memory (or DRAM), for short-term “volatile” storage

- Retains data only while computer is powered on
- Size measured in GB, usually 4 or 8GB for a laptop these days

Disk (HDD or SSD), for long-term persistent storage

- Retains data even when computer is powered off
- Size measured in GB (1 billion bytes), or even TB (1000 GB)

Power Supply and/or Battery

Input/Output Devices

- Integrated
- Connected via USB, Bluetooth, etc.

SOFTWARE: OPERATING SYSTEMS, APPLICATIONS, APPS

- Operating systems
 - Windows
 - macOS,
 - Linux
 - Chrome
 - FreeBSD
 - iOS
 - Android
- Applications and apps
 - web browsers
 - social media
 - video streaming
 - many more ...

SOFTWARE BASICS

Software tells hardware what to do

- The processor is the “programmable” part of the hardware

Processor understands primitive “machine language”

- Load from memory or store to memory or device
- Simple arithmetic operations like add, subtract, multiply, divide
- Branch (jump to a different location)
- Conditional Branch (jump to a different location if a certain condition is true)

With these simple primitives, you can implement any software application!

Most people program computers using higher level languages

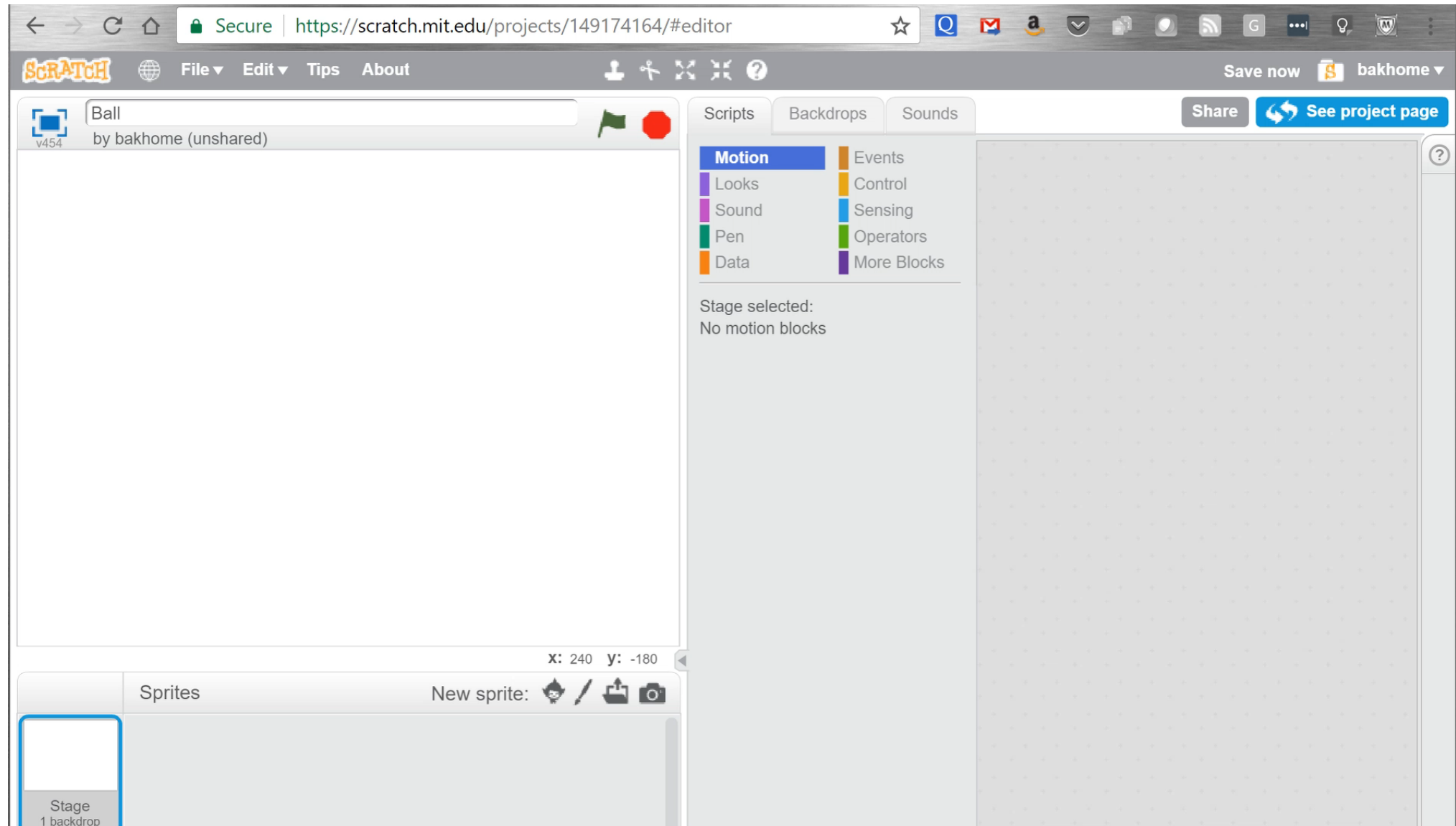
HIGHER LEVEL PROGRAMMING LANGUAGES

Language	Where it's used
Scratch	Visual programming language used for teaching programming
Python	Scripting language for web programming and desktop applications
Java	One of the most popular programming languages, used for Android Apps and client-side web applications
Swift	Used for programming iOS Apps
C	Used for operating systems, server-side applications
C++, C#	
JavaScript	
HTML/CSS	“Hyper Text Markup Language” used for creating web pages, “Cascading Style Sheets”

Not enough time today to teach you a language
Will give examples from Scratch, Python and Java

SCRATCH

VISUAL PROGRAMMING

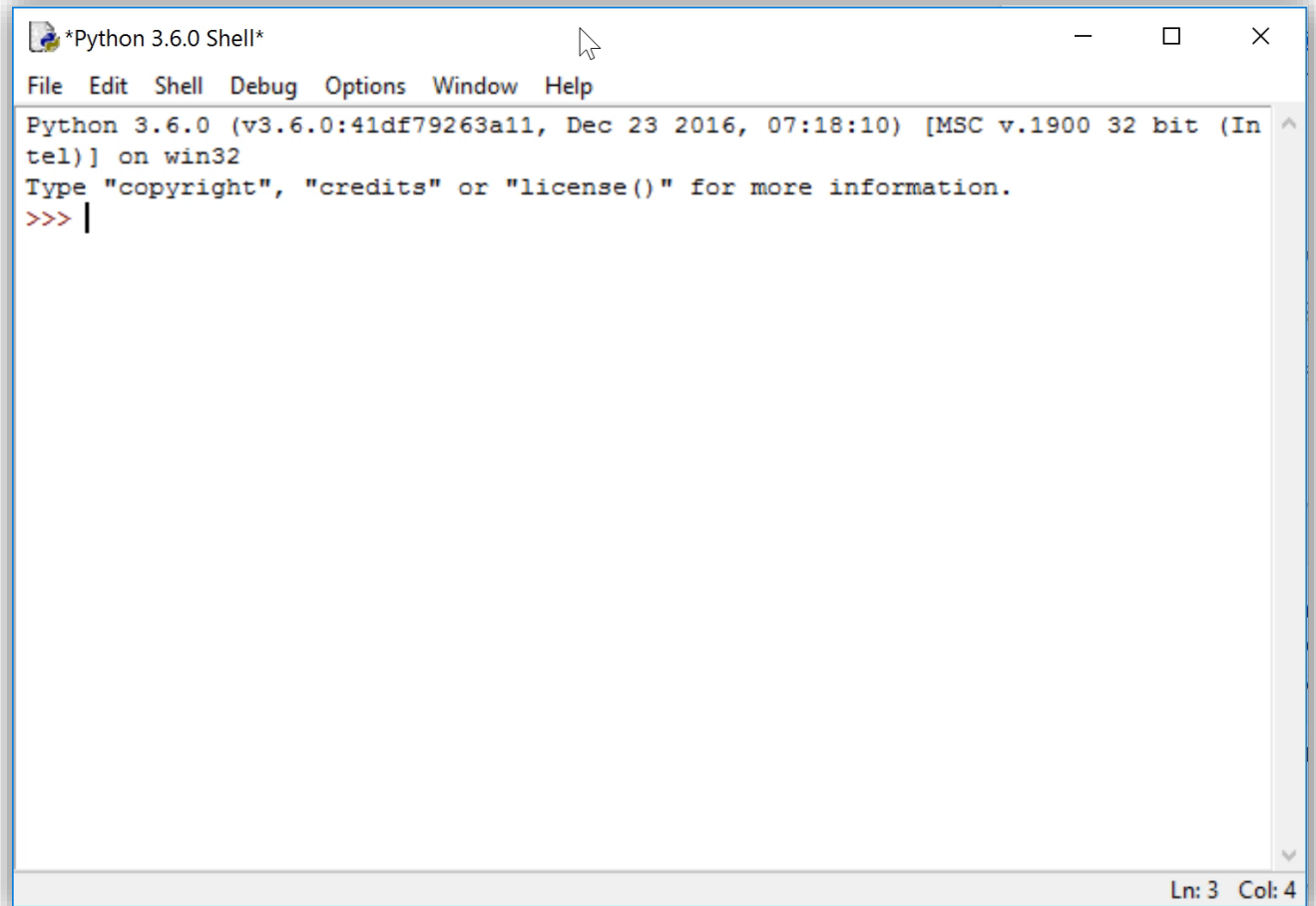


PYTHON |

PYTHON – HELLO WORLD

Download Python from
<https://www.python.org/downloads/>

First program: print “Hello,
world!” to the screen uses the
“print()” function



```
*Python 3.6.0 Shell*
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> |
```

Ln: 3 Col: 4

SIMPLE ARITHMETIC AND MODULES

Basic Arithmetic

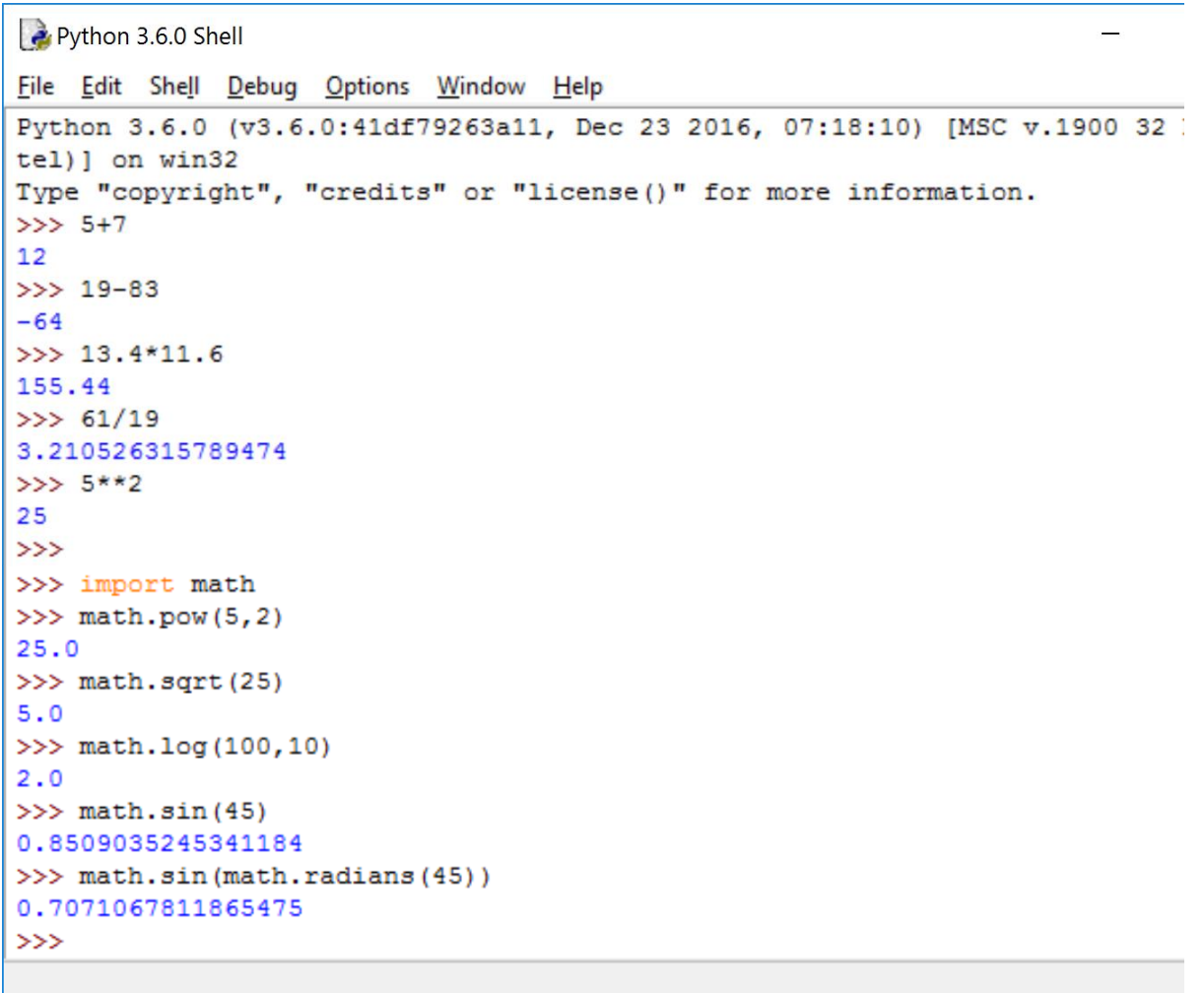
- Addition, Subtraction
- Multiplication, Division
- Power

Math Module

- Provides higher level “functions” and constants
- Powers and Logarithmic functions
- Trigonometric functions
- Constants: pi, e

Other modules

- datetime
- random
- Many, many more!

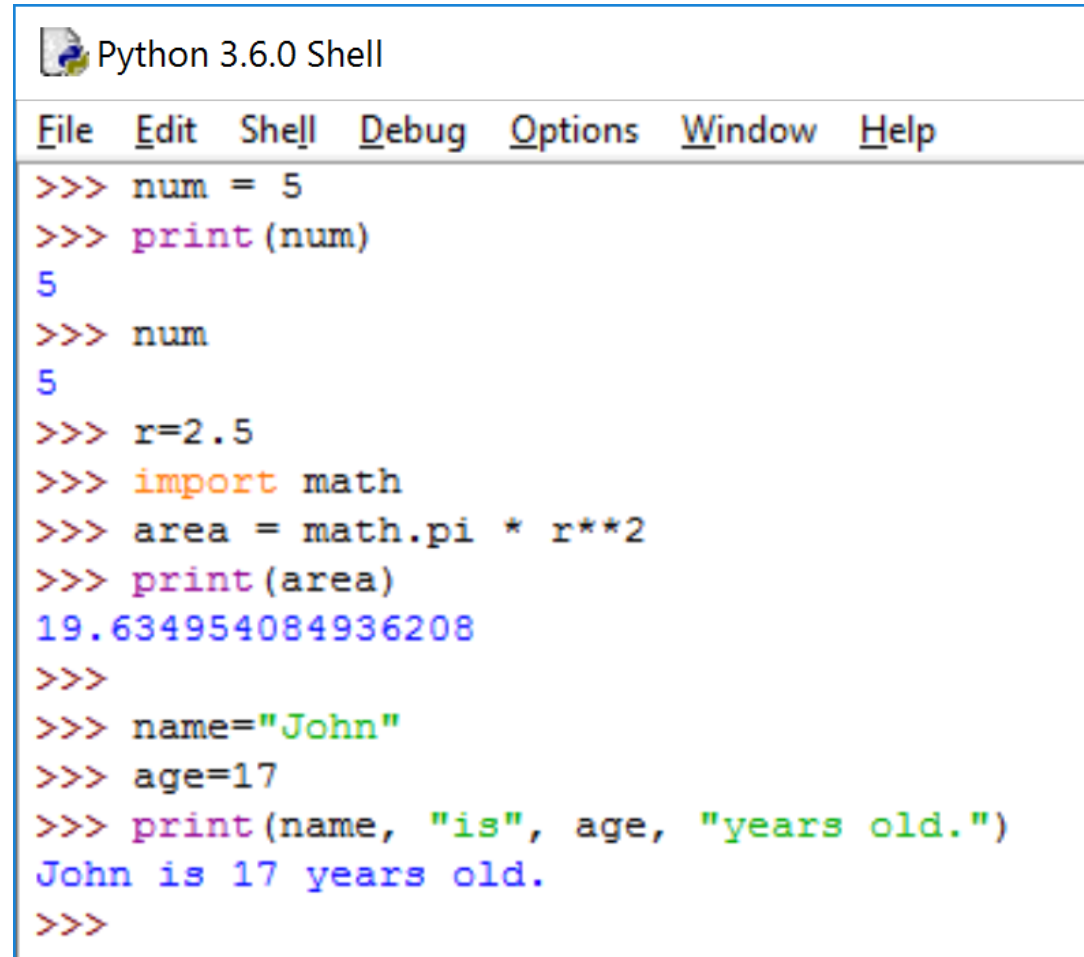
A screenshot of a Python 3.6.0 Shell window. The window has a title bar that says "Python 3.6.0 Shell" and a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main area shows the following text:

```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 :  
tel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>> 5+7  
12  
>>> 19-83  
-64  
>>> 13.4*11.6  
155.44  
>>> 61/19  
3.210526315789474  
>>> 5**2  
25  
>>>  
>>> import math  
>>> math.pow(5,2)  
25.0  
>>> math.sqrt(25)  
5.0  
>>> math.log(100,10)  
2.0  
>>> math.sin(45)  
0.8509035245341184  
>>> math.sin(math.radians(45))  
0.7071067811865475  
>>>
```

VARIABLES

A “variable” is a place to store a value

- Numbers
 - Integers
 - Floating Point numbers
- Strings
 - Used to store text – a name, word, sentence, paragraph
 - Surrounded by ‘single’ or “double” quotes



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
>>> num = 5
>>> print(num)
5
>>> num
5
>>> r=2.5
>>> import math
>>> area = math.pi * r**2
>>> print(area)
19.634954084936208
>>>
>>> name="John"
>>> age=17
>>> print(name, "is", age, "years old.")
John is 17 years old.
>>>
```

LOOPS

While Loop

- Example: Print out the square of each number from 0 to 10

```
>>> num = 0
>>> while (num <= 10):
    print(num, "squared =", num**2)
    num = num + 1
```

```
0 squared = 0
1 squared = 1
2 squared = 4
3 squared = 9
4 squared = 16
5 squared = 25
6 squared = 36
7 squared = 49
8 squared = 64
9 squared = 81
10 squared = 100
>>> |
```

INTERLUDE: ALGORITHM

An algorithm is a set of steps to perform

- Like a recipe for completing a task

Example: Compute Age

- **Inputs**

- Year of birth: Integer
- Whether birthday already celebrated this year

- **Outputs**

- age: Integer

- **“Pseudo code”**

- Get current year
- $\text{age} = (\text{current year}) - (\text{year of birth})$
- if (birthday NOT already celebrated this year) then decrement age by 1

COMPUTING AGE IN PYTHON – STEP 1

Get Inputs from User

Use the “input()” function to get input (birthYear) from the user

Get currentYear

Import datetime module

Call date.today() function to get current date

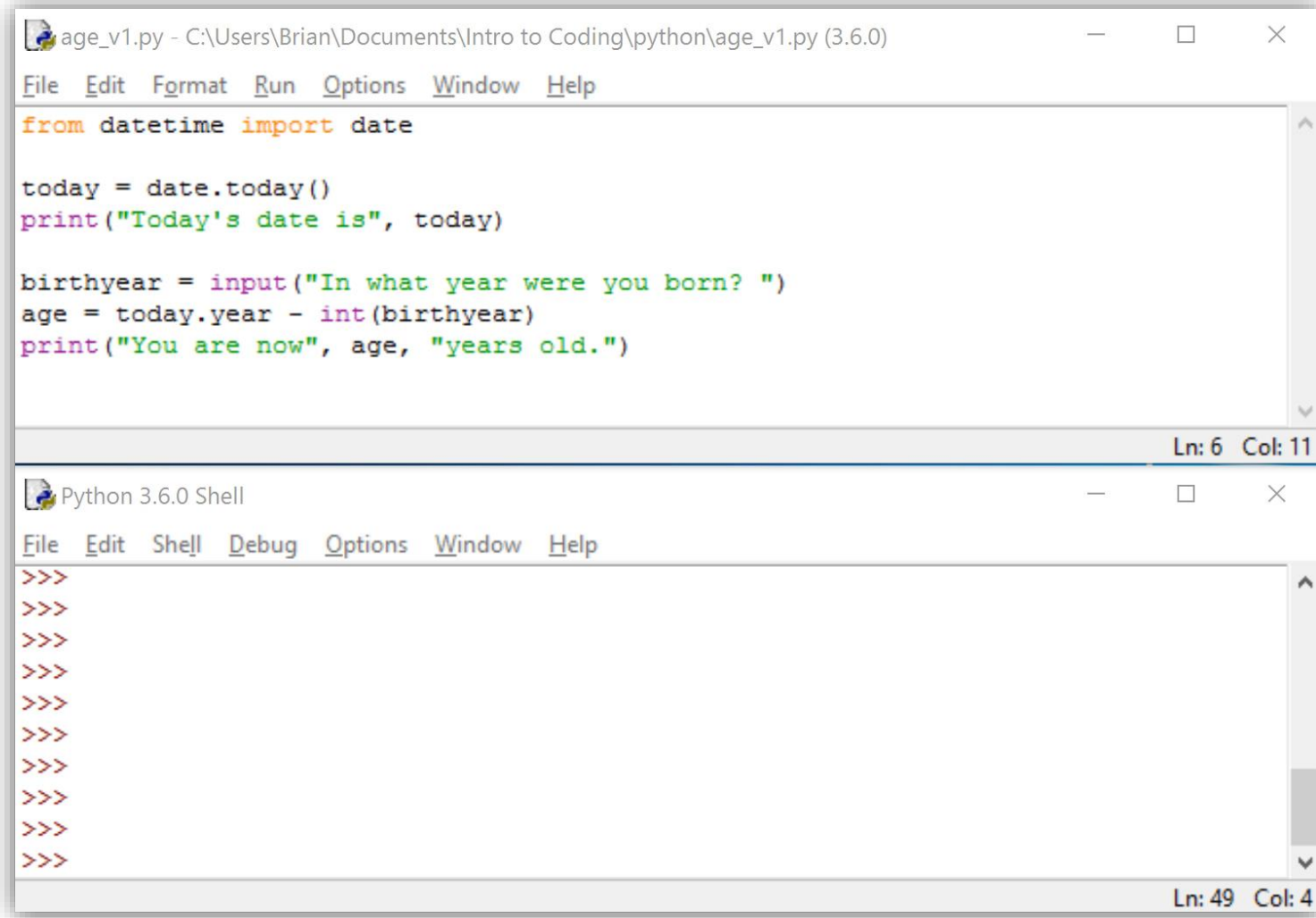
Use .year field to determine year

Compute age

Use int() function to convert a string to an integer

```
>>> birthYear = input("In what year were you born? ")
In what year were you born? 2000
>>> birthYear
'2000'
>>>
>>> from datetime import date
>>> date.today()
datetime.date(2017, 3, 5)
>>> currentYear = date.today().year
>>>
>>> age = currentYear - birthYear
Traceback (most recent call last):
  File "<pyshell#65>", line 1, in <module>
    age = currentYear - birthYear
TypeError: unsupported operand type(s) for -: 'int' and 'str'
>>> age = currentYear - int(birthYear)
>>> age
17
>>>
```

USING SEPARATE WINDOWS FOR EDITING AND RUNNING THE CODE



CONDITIONS AND FUNCTIONS

IF statements

```
answer = input("Have you celebrated your birthday yet this year? ")  
  
if (answer.lower() == 'no'):  
    age = age - 1
```

Defining a Function

```
# Function to compute age  
# birthYear: integer specifying year of birth  
# celebratedBirthday: True if birthday has been celebrated this year  
def computeAge(birthYear, celebratedBirthday):  
    age = date.today().year - birthYear  
    if (not celebratedBirthday):  
        age = age - 1  
    return age
```

PUTTING IT ALL TOGETHER

```
age_v5.py - C:/Users/Brian/Documents/Intro to Coding/python/age_v5.py (3.6.0)
File Edit Format Run Options Window Help

from datetime import date

# Define a function to compute age
# This takes two input parameters:
# - birthYear: integer specifying year of birth
# - celebratedBirthday: True if birthday has been celebrated this year
def computeAge(birthYear, celebratedBirthday):
    age = date.today().year - birthYear
    if (not celebratedBirthday):
        age = age - 1
    return age

# Main program starts here
birthyear = input("In what year were you born? ")
answer = input("Have you celebrated your birthday yet this year? ")

# Now call the computeAge function above
age = computeAge(int(birthyear), (answer.lower() == 'yes'))

# Finally, print the result to the screen
print("You are now", age, "years old.")

Ln: 18 Col: 54
```

```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help

>>>
===== RESTART: C:/Users/Brian/Documents/Intro to Coding/python/age_v5.py =====
In what year were you born? 2000
Have you celebrated your birthday yet this year? no
You are now 16 years old.
>>>

Ln: 34 Col: 4
```

ANDROID APP DEVELOPMENT WITH JAVA

INTERLUDE: ALGORITHM VARIANT: CALCULATE AGE

Inputs

- birthyear: Integer
- birthMonth: Integer
- birthDayOfMonth: Integer

Outputs

- age: Integer
- message: String (says “Happy Birthday” on birthday)

“Pseudo code”

- Get currentYear, currentMonth, currentDayOfMonth
- $age = currentYear - birthYear$
- if (currentMonth < birthMonth) then
 - $age = age - 1$
- else
 - if (currentMonth == birthMonth) then
 - if (currentDayOfMonth < birthDayOfMonth) then
 - $age = age - 1$
 - else if (currentDayOfMonth == birthDayOfMonth) then
 - message = “Happy birthday!”

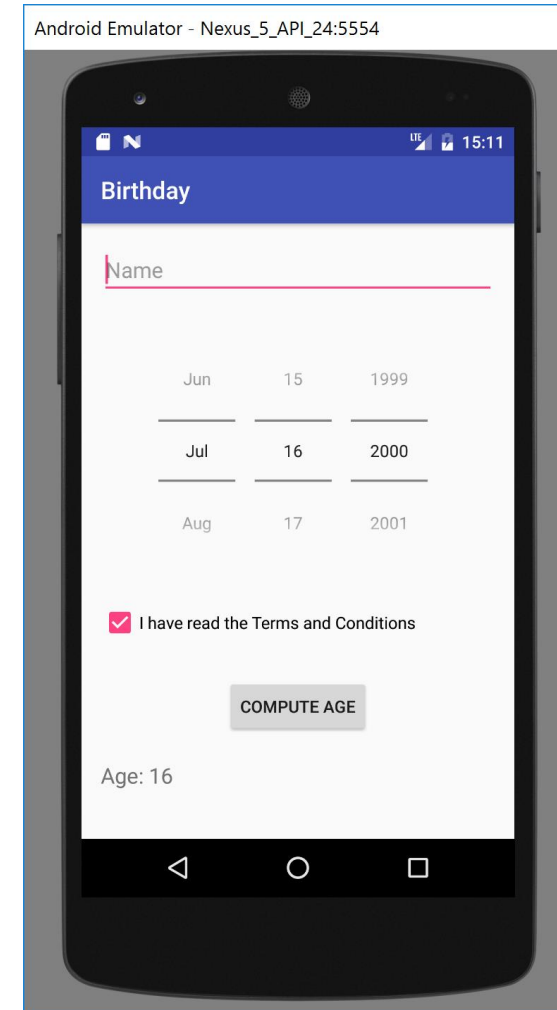
PROGRAMMING GRAPHICAL USER INTERFACES (GUIs)

Elements

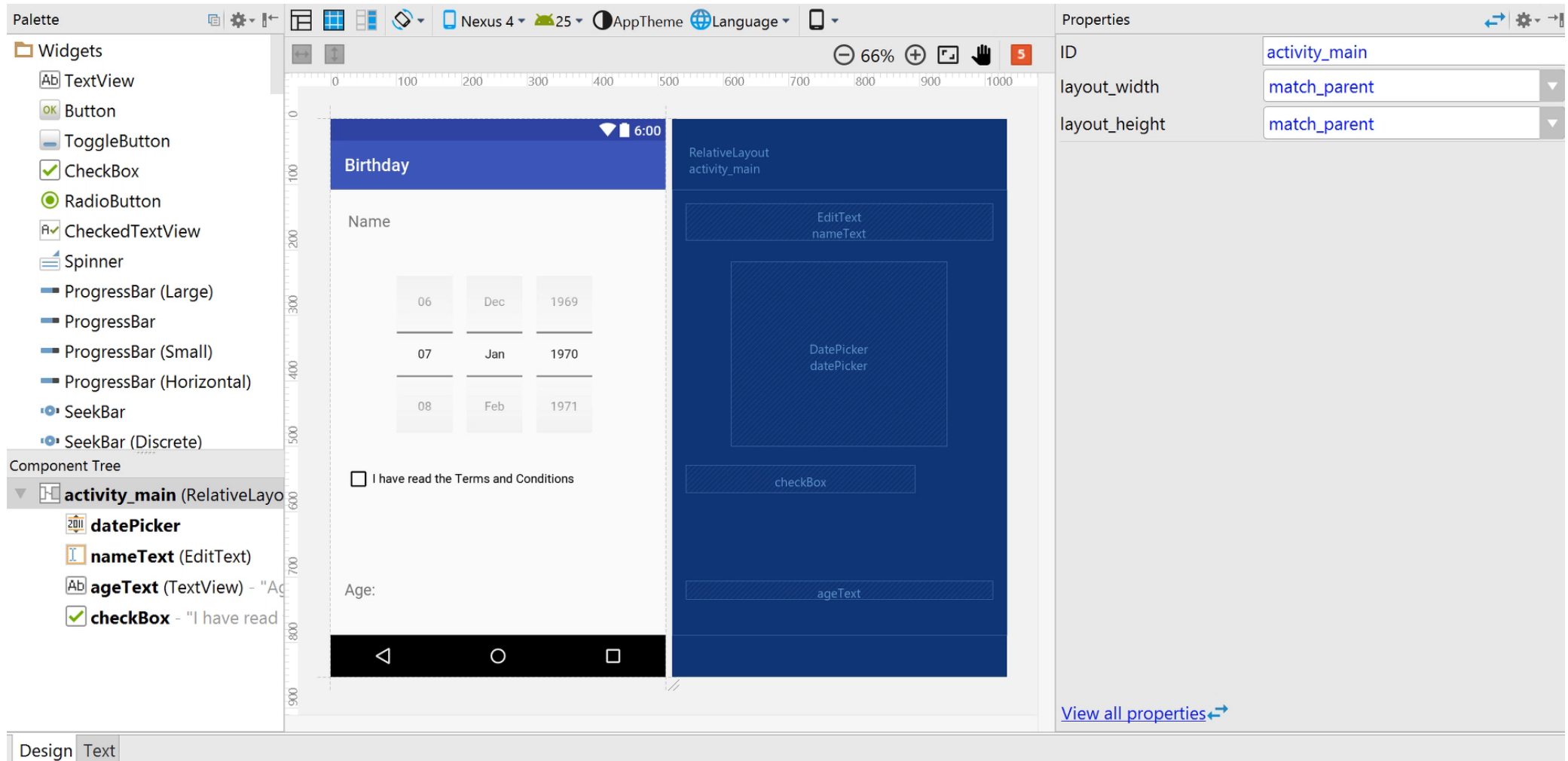
- EditText (for text input)
- DatePicker
- Checkbox
- Button
- TextView (for text output)

Each has properties

- Appearance
- Text
- Function to call on specified events



DRAWING THE USER INTERFACE



CODE TO HANDLE BUTTON CLICK

```
public void computeAge(View view) {
    DatePicker datePicker = (DatePicker) findViewById(R.id.datePicker);
    EditText nameText = (EditText) findViewById(R.id.nameText);
    TextView ageText = (TextView) findViewById(R.id.ageText);

    // get date from date picker
    int birthYear = datePicker.getYear();
    int birthMonth = datePicker.getMonth();
    int birthDayOfMonth = datePicker.getDayOfMonth();

    // get current date
    Calendar cal = Calendar.getInstance();
    int currentYear = cal.get(Calendar.YEAR);
    int currentMonth = cal.get(Calendar.MONTH);
    int currentDayOfMonth = cal.get(Calendar.DAY_OF_MONTH);

    int age = currentYear - birthYear;
    String happyBirthayMessage = "";

    if (currentMonth < birthMonth) {
        age--;
    }
    else {
        if (currentMonth == birthMonth) {
            if (currentDayOfMonth < birthDayOfMonth) {
                age--;
            }
            else {
                if (currentDayOfMonth == birthDayOfMonth) {
                    happyBirthayMessage = "\nHappy birthday, " + nameText.getText() + "!";
                }
            }
        }
    }

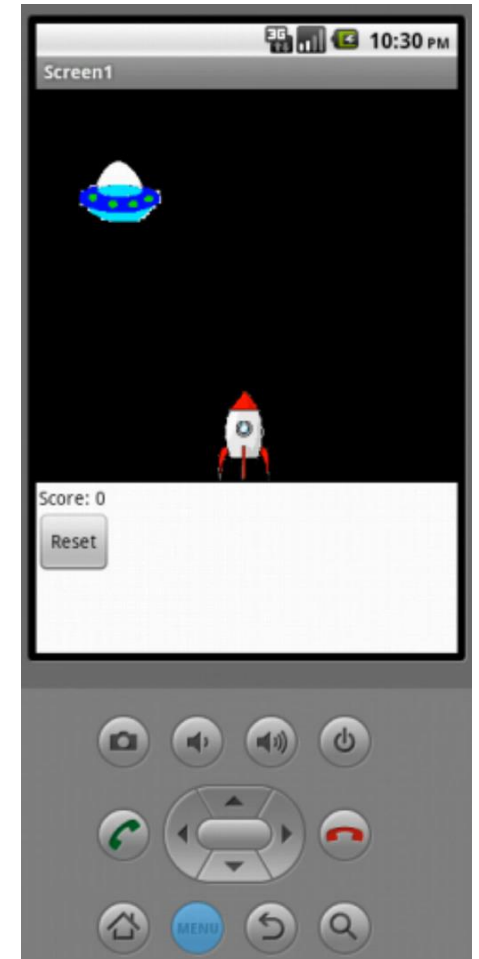
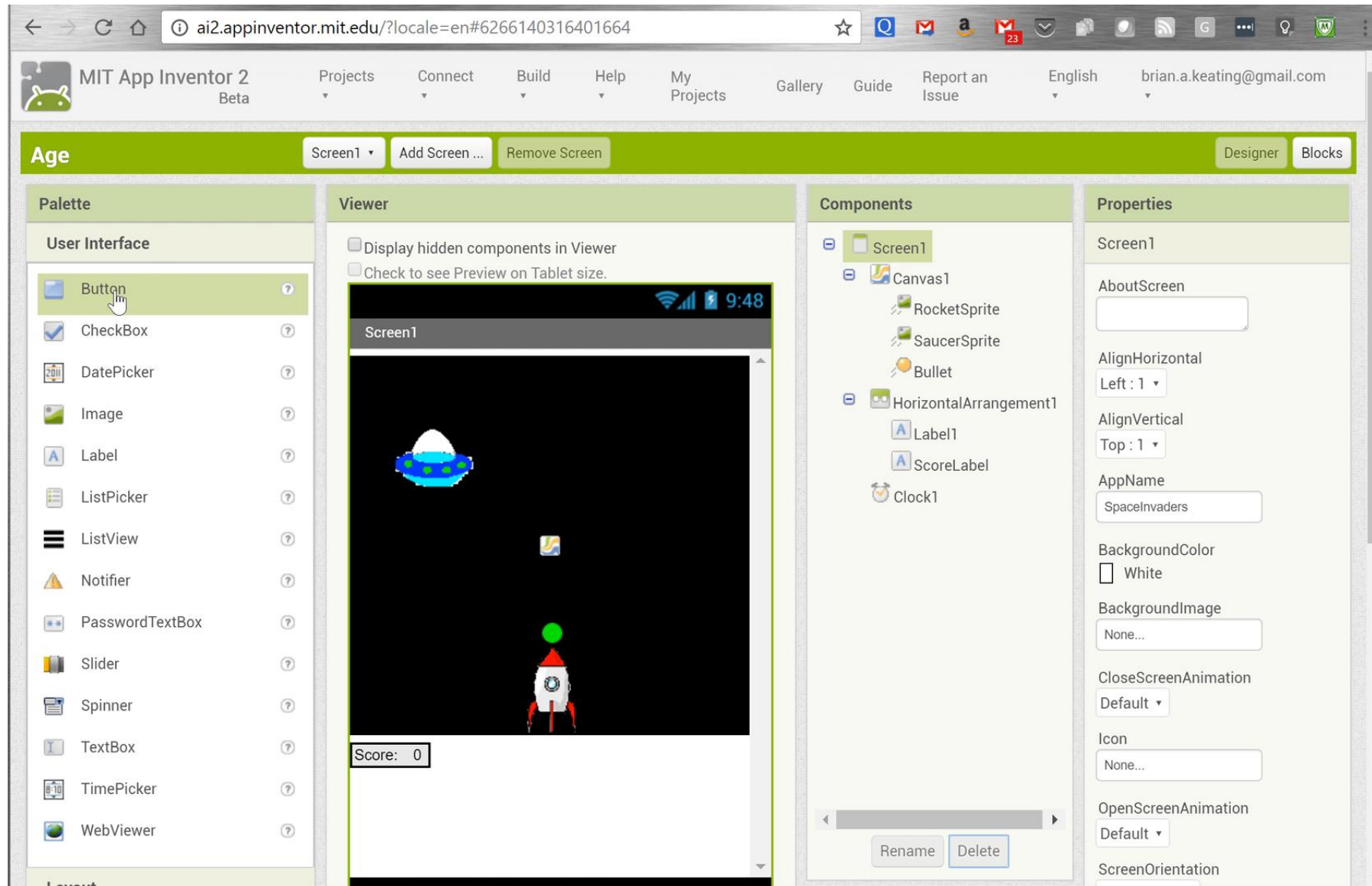
    ageText.setText("Age: " + Integer.toString(age) + happyBirthayMessage);
}
```



ANDROID APP DEVELOPMENT WITH APP INVENTOR



VISUAL PROGRAMMING FOR APPS!



INTERNET OF THINGS DEVELOPMENT



CODING FOR THE INTERNET OF THINGS

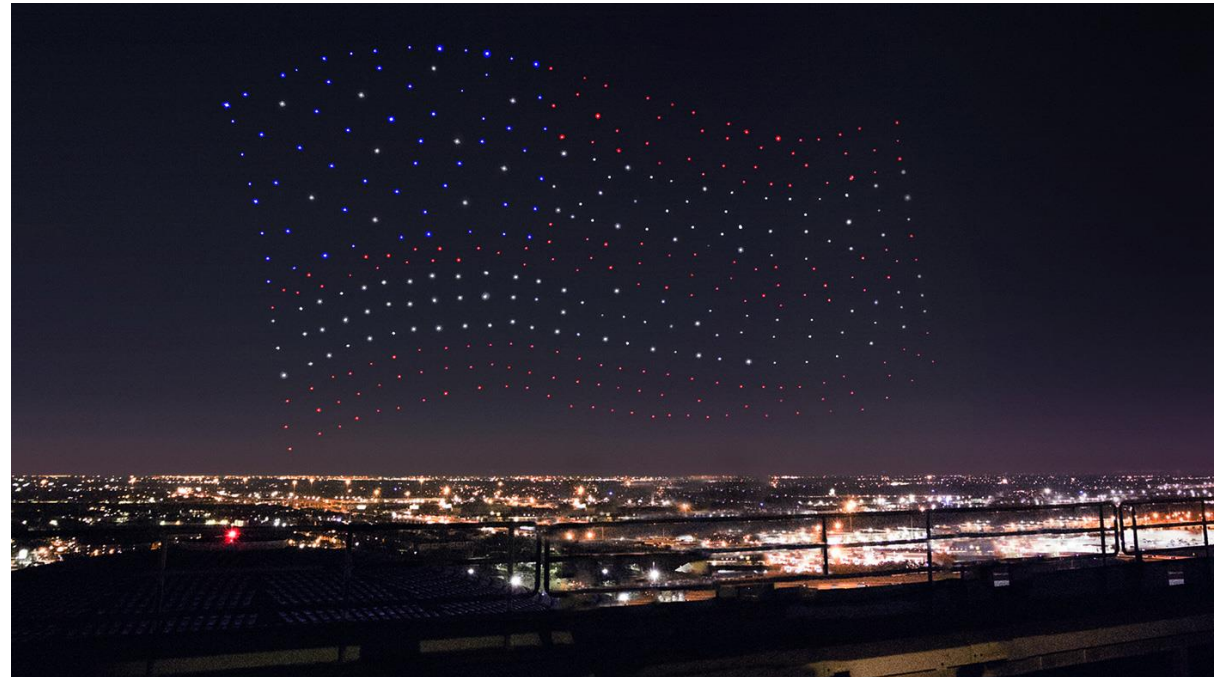
Get going on developing IoT devices with a starter-level development board and add-on kit (for sensors and actuators)



&



More advanced ...

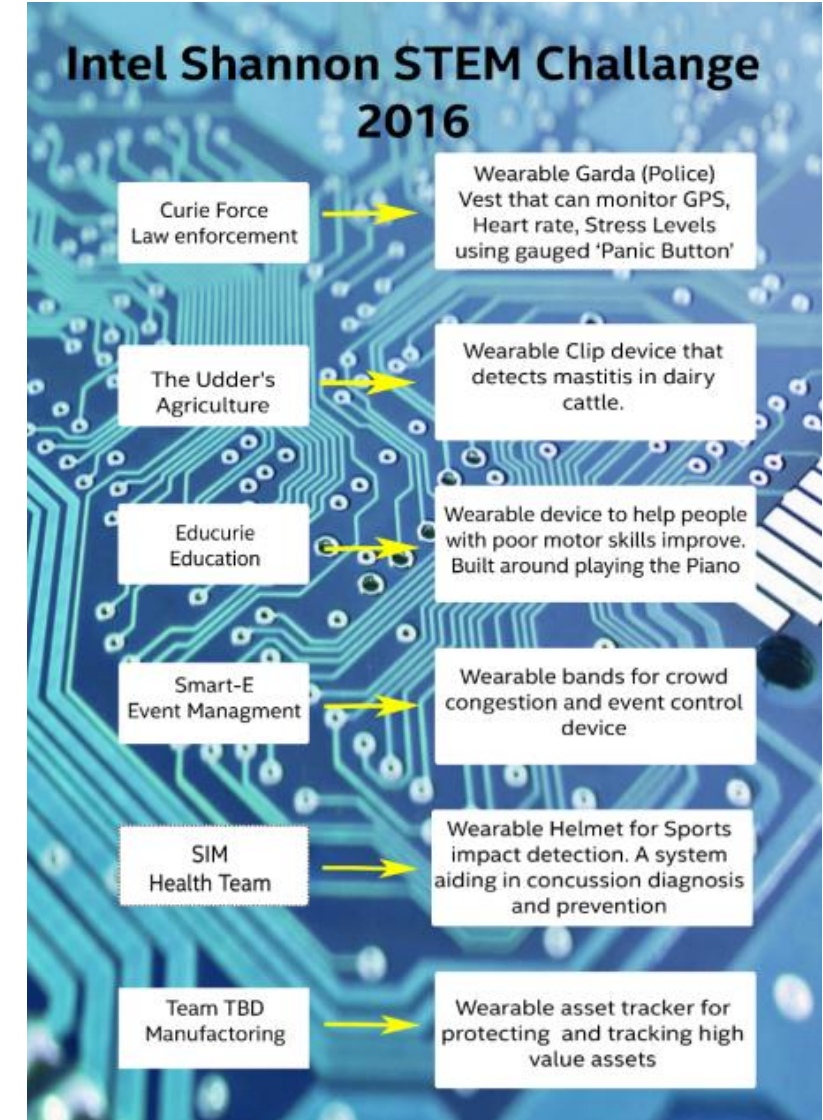


Intel Drone show at Superbowl 2017

<https://www.youtube.com/watch?v=OyPCNvx9hVQ>

CODING AT INTEL: THE STEM CHALLENGE

Intern challenge to develop IoT products with an Intel board development kit



FOR MORE INFORMATION |

ONLINE COURSES (BEGINNERS)

Code.org

- <https://studio.code.org> – courses for beginners
- <https://code.org/learn> – hour-of-code tutorials for beginners

Code Academy

- <https://www.codecademy.com/learn/python> – learn Python
- <https://www.codecademy.com/learn/javascript> – learn Javascript
- <https://www.codecademy.com/learn/learn-html-css> – learn HTML and CSS
- <https://www.codecademy.com/learn/learn-java> – learn Java programming

Khan Academy

- <https://www.khanacademy.org/computing/computer-programming/programming> – using JavaScript
- <https://www.khanacademy.org/computing/computer-science/algorithms> – computer algorithms
- <https://www.khanacademy.org/hourofcode> – hour-of-code tutorials

ONLINE COURSES (MORE ADVANCED)

Coursera

NOTE: some Coursera courses may require fee

- <https://www.coursera.org/learn/intro-programming> – using Scratch
- <https://www.coursera.org/learn/python> – Python course
- <https://www.coursera.org/specializations/java-programming> – Java programming course
- <https://www.coursera.org/learn/game-programming> – games programming using C#
- <https://www.coursera.org/specializations/iot> – Intro to Internet of Things programming
- <https://www.coursera.org/learn/arduino-platform> – IoT programming in C on Intel Genuino

Lynda.com

- <https://www.lynda.com/C-tutorials/C-Essential-Training/164457-2.html> – C programming
- <https://www.lynda.com/Android-tutorials/Up-Running-Java-Applications/94344-2.html> – Java

CODING TOOLS

Scratch

- <http://scratch.mit.edu> – online tool for coding in Scratch

Python

- <https://www.python.org/downloads> – download and install Python
 - Recommend latest 3.x version; some online courses (e.g. code-academy) use Python 2.x

Android App Development

- <http://appinventor.mit.edu> – App Development using Blocks
- <https://developer.android.com/studio/index.html> – Android Studio Development

iOS App Development (for iPhone and iPad)

- <http://www.apple.com/ie/swift/playgrounds> – Swift Playgrounds iPad app
- <https://developer.apple.com/xcode> – Development Environment for iPhone, iPad

INTERNET-OF-THINGS DEVELOPMENT

Intel Maker Movement

- <http://www.intel.ie/content/www/ie/en/do-it-yourself/maker.html>
- <http://www.intel.ie/content/www/ie/en/do-it-yourself/genuino-101.html>

Intel Genuino 101 development board

- <https://www.arduino.cc/en/Guide/Arduino101>
- <https://software.intel.com/en-us/articles/fun-with-the-arduino-101-genuino-101>

NOTE: Intel starter development board known as Genuino101 in Europe, and as Arduino101 in America.

Can purchase development board and sensor/actuator kit at Maplin, Childers Rd Retail Park, Limerick or online at <http://www.mouser.ie>

“Intel Genuino101 Development Board” Manufacturer part number: GBX00005

“Grove Starter kit for Genuino101” Manufacturer part number: 110020109

CODING CLUBS

Coder Dojo runs coding clubs in Limerick

- <http://www.coderdojomidwest.com>
- Saturdays 2–4pm in LIT
- Wednesdays 7–9pm in Mary I College

VIDEOS TO WATCH

Computer Coding – what most schools don't teach

(featuring Facebook founder Mark Zuckerberg, Will.I.Am and many more)

- <https://www.youtube.com/watch?v=nPblG6ceqOs>

Intel drone technology at Super-bowl half-time show (example of internet-of-things)

(featuring Lady Gaga)

- <https://www.youtube.com/watch?v=OyPCNvx9hVQ>
- <https://www.youtube.com/watch?v=mk7TsDNuhs0>