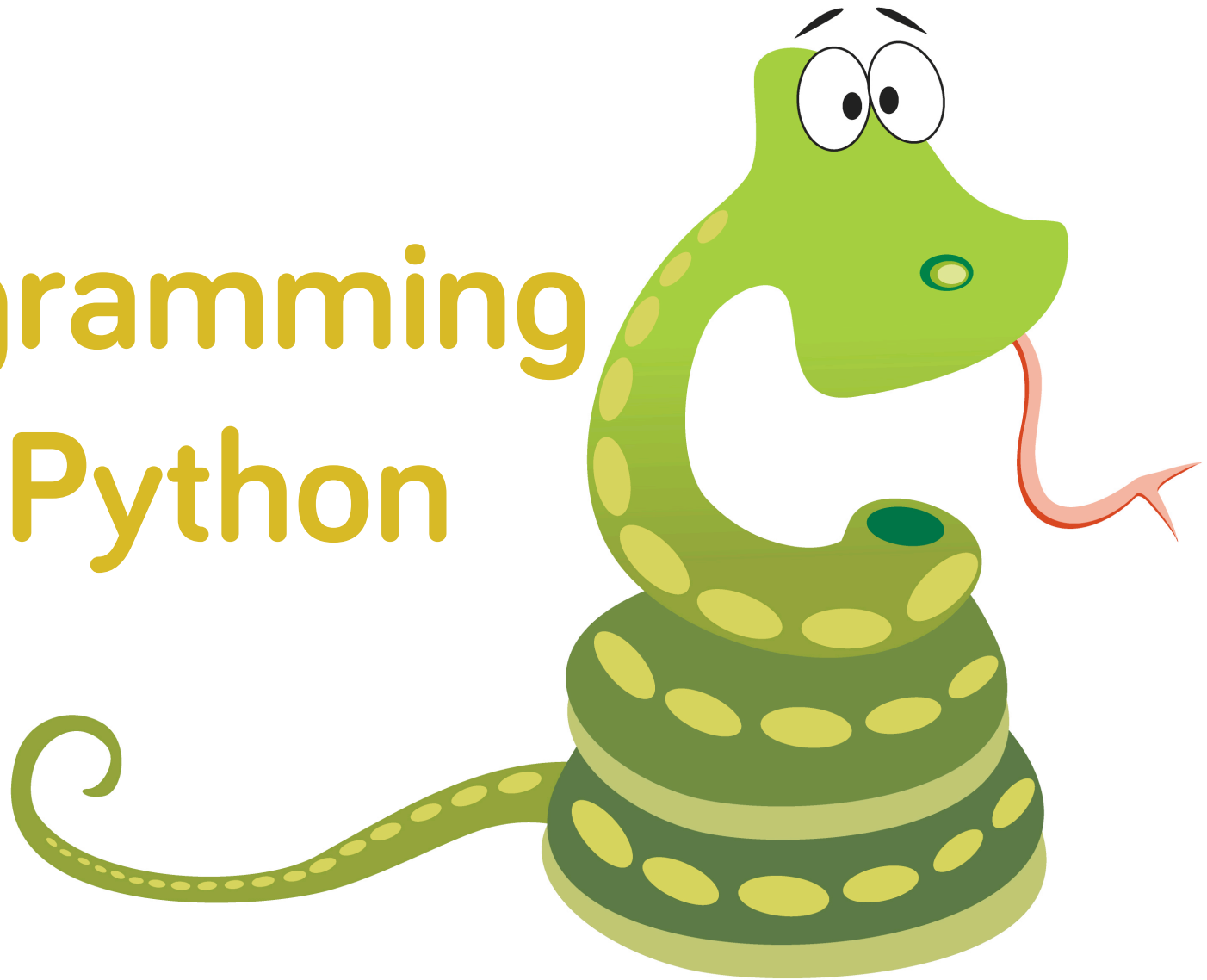


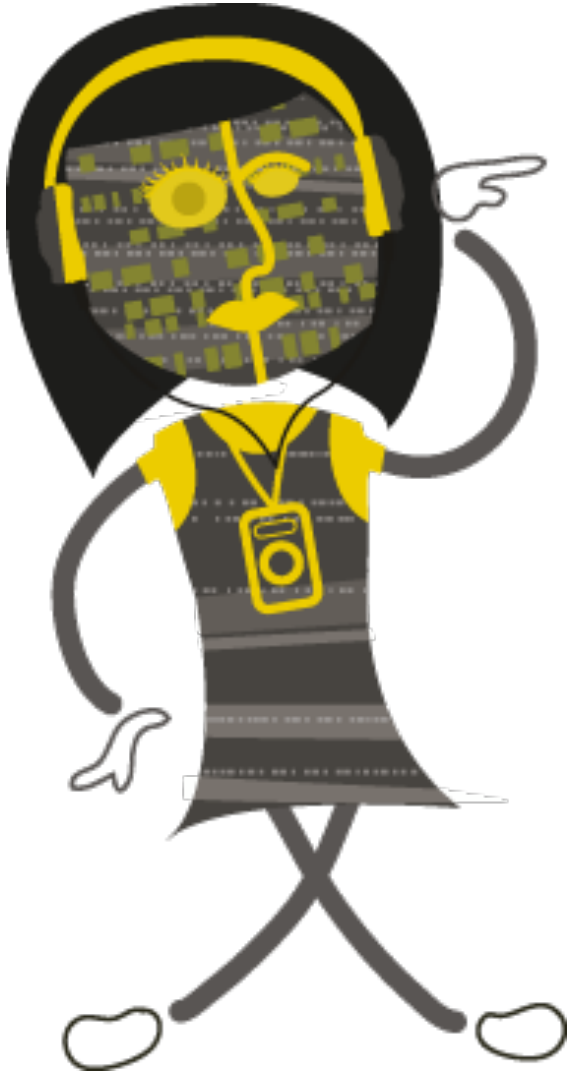
technocamps

www.technocamps.com



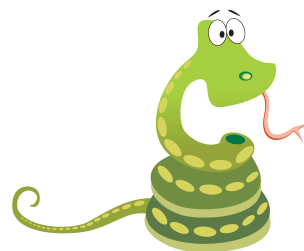
Programming in Python





Okay...so we won't
be learning about
snakes today...

What will we be learning?



What is Computer Science?

Has anyone done any type of programming before?



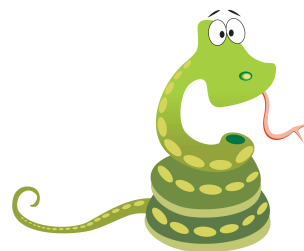
Let's practice our instruction skills...



Task one: Drawing

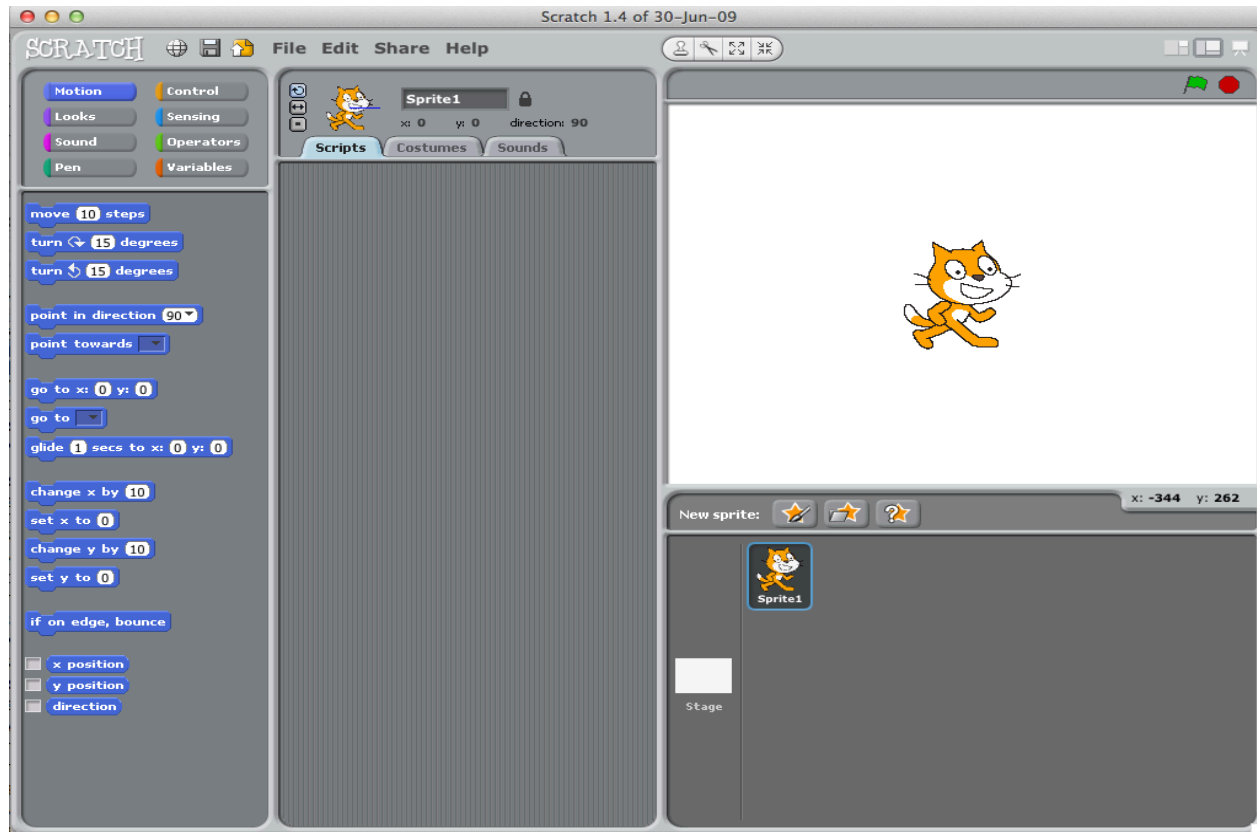
Task two: Programming a robot

Task three: Dance



SCRATCH

RE-CAP



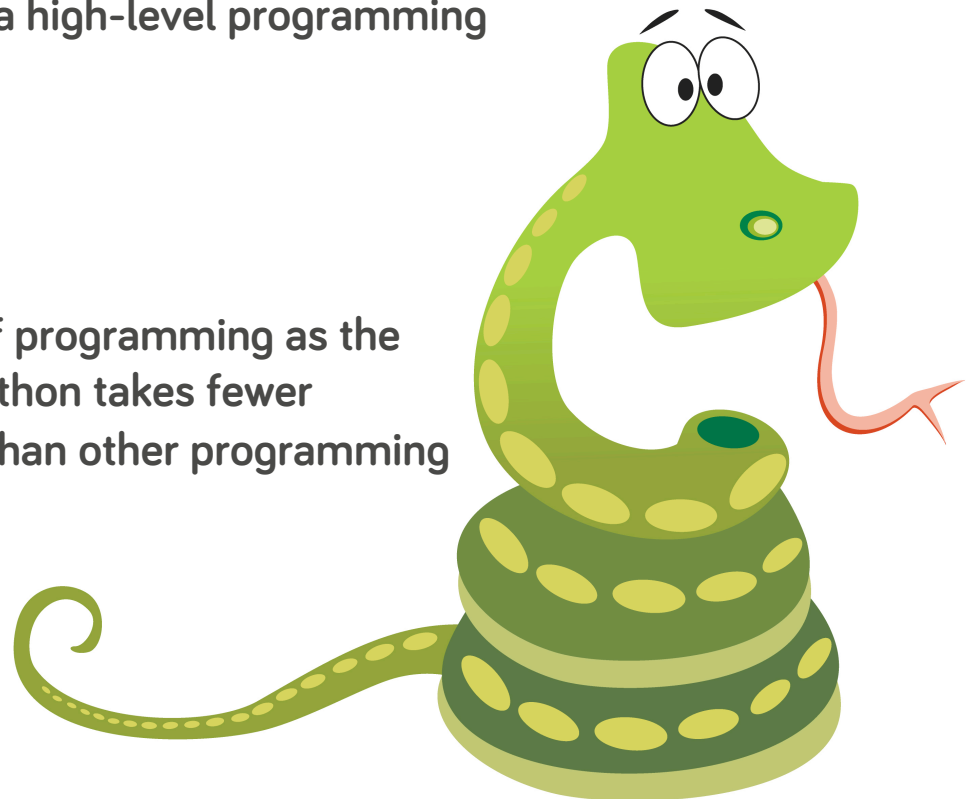
An introduction to Python

What is Python?

Python is a programming language that is a high-level programming language.

Why learn Python?

Python is used a lot within the educating of programming as the code is very readable and easy to learn. Python takes fewer lines of code to express a coding concept than other programming languages.



Python

Python is a programming language that looks like this:

```
print ("Hello, welcome to Technocamps!")

response = raw_input("Would you like to learn Python?")
if response == "yes":
    print ("Great! Have a look on the interactive zone to get started!")
else:
    print ("No problem, how about some of our other workshops?")
```

We use quote marks
to input strings!

Using colons to end
statements

See the use of
indentation to show what
code is part of the if
statement!

Python is a language that is easy to read and is a great language
for beginners!

Today...

We will be trying out some Python ourselves, creating a text-based puzzle game.

We will be learning:

- How to confidently code using Python
- What “loops” are and how they can be implemented
- How to recognise Python and identify what the code means
- How to implement conditional statements
- How to create our own variables
- How to create our own functions and how they can be efficiently used

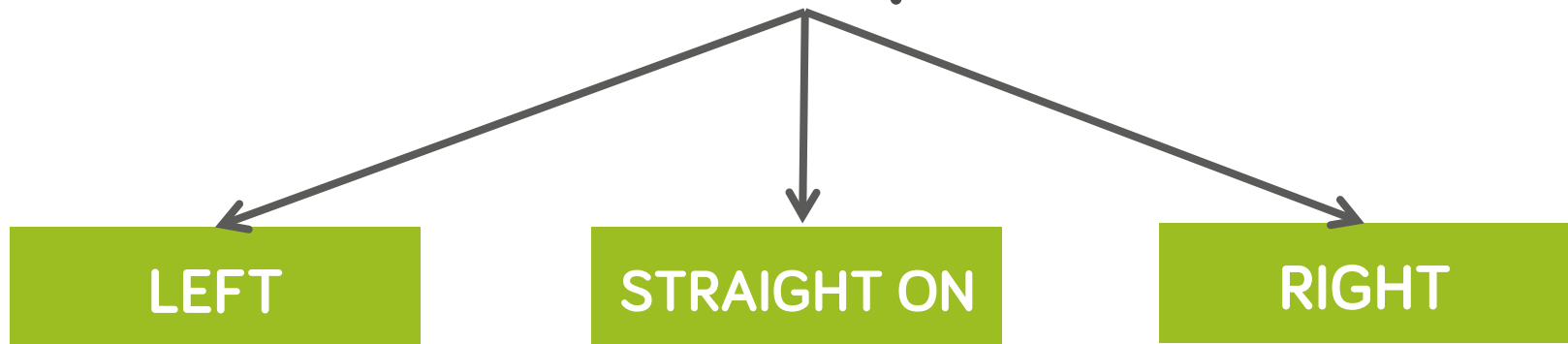
Let's start by designing our game...



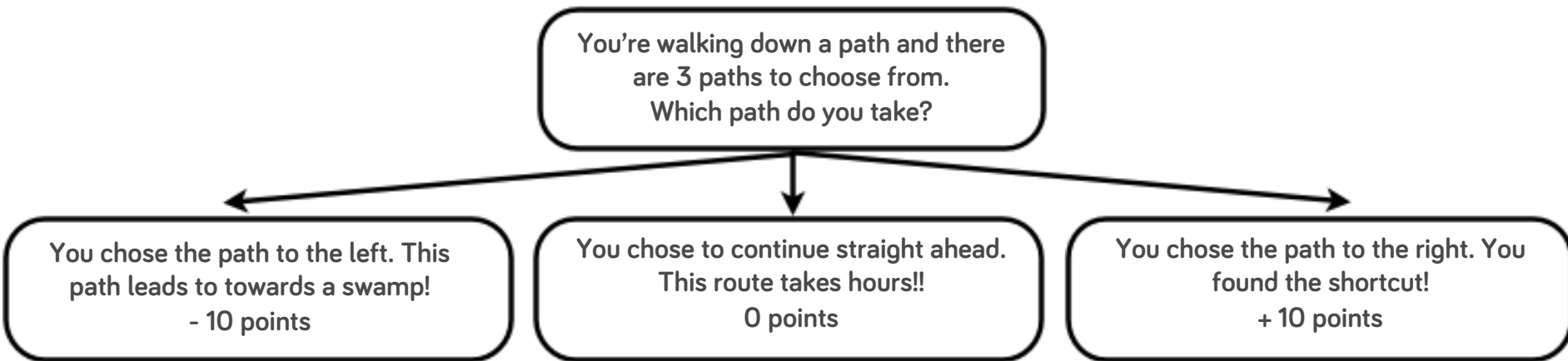
Designing your game #1

You're walking down a path and there
are 3 paths to choose from.
Which path do you take?

Choose a path...



Designing your game #2



Using the blank templates, design and expand on your own puzzle quiz.

Notes:

Each question must be able to be answer with written user input.
You don't need to have 3 options, you could even have "True or False" if you wanted.

Print

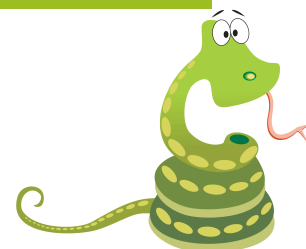
When learning a new programming language, or learning to program for the first time traditionally the first task to achieve is to output text (usually “Hello world”).

CODE: `print ("Hello world")`

OUTPUT: Hello world

The above command will output (print) whatever is between the “ and “.

This is a good way to provide any users information. It can also be used later on outputting questions to collect user data on input when we use `raw_input()`.



Variables

A variable is a part of the program that can store data. This storage can be named anything, what's important is the contents of this storage. Fortunately with Python, we do not need to declare what type of content it is e.g. integer, string etc.

CODE : `output_text = "Hello world"`
`Print output_text`

OUTPUT : Hello world



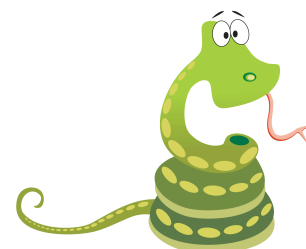
Raw_input

Raw_input is one of many built in functions for Python, functions that have already been defined. Using raw_input allows us to ask the user a question and collect an input response. E.g. How old are you? The user would answer and whatever they typed would be saved in the variable called “answer”.

CODE: `answer = raw_input("How old are you?")`

OUTPUT: How old are you? **input required**

How old are you?



Print and raw_input combined

CODE:

```
answer = raw_input("How old are you?")  
print "You are " + answer + " years old"
```

OUTPUT:

```
How old are you?  
  
*user types 17*  
  
You are 17 years old
```

How old are you?

17|

You are 17 years old

Loops

Sometimes we want to repeat instructions, or get the program to execute a line of code or numerous lines.

For loop

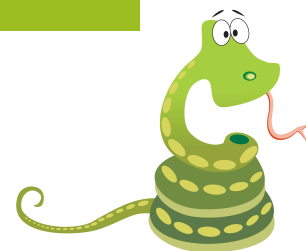


This statement enables us to set the amount of times we want to execute the code within the statement (indented).

While loop



Repeats a line/lines of code, while a certain condition is true. It tests if the condition is true and executes the code within the statement. If the condition becomes false, this code is not executed.



Loops: FOR

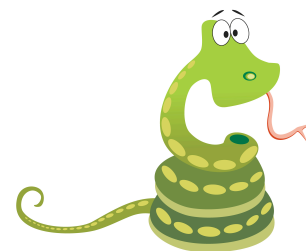
CODE:

```
for letters in "I love Python":  
    print letters
```

OUTPUT:

```
I  
l  
o  
v  
e  
  
P  
y  
t  
h  
o  
n
```

As you can see, letters is the variable being used here. If there are no more letters left in the sequence, the for loop is terminated and no more letters are printed as output.



Loops: WHILE

CODE :

```
counter = 0
while (counter < 10):
    print "The counter is " + counter
    counter += 1
print ("The End.")
```

OUTPUT :

0 As you can see, counter is the variable being used here. It
1 was created on the first line and assigned the value 0.
2 It will only execute the lines within the while loop, as long as
3 the counter variable value is less than 10. Each time this is
4 true, the variable counter adds 1. As you can see, the number
5 10 is not printed, only values LESS THAN 10.

6
7 This can be expanded to become an WHILE...ELSE statement that can
8 execute lines of code when the condition is no longer true.
9



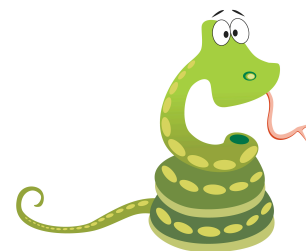
Conditional Statements

An if statement can be used to state “if a certain condition occurs, execute the following code. Else, if anything other than that condition occurs, execute this code instead”. This statement can also be expanded to become an IF, ELIF & ELSE statement, so 2 different conditions can be described within IF and ELIF. The IF statements can also combine and nest loops and/or built in functions of Python’s, such as raw_input.

CODE :

```
my_var = 37
if my_var < 50:
    print "It is less than 50!"
else:
    print "It is more than 50!"
```

OUTPUT : It is less than 50!



Functions

A function is a block of re-usable code. It could be coded and re-used numerous times throughout the program you are creating. Python does have some built-in functions that have already been declared, but you can create your own using code similar to below.

CODE :

```
def print_name(x):  
    print "Your name is" + x
```

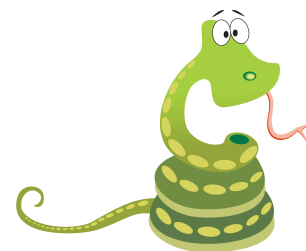
This is how you call the function, putting the input between the (and).



```
print_name("Rob")
```

OUTPUT :

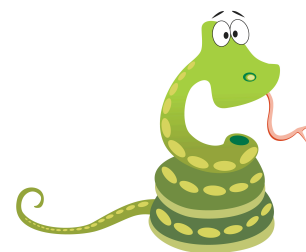
```
Your name is Rob
```



Syntax

The programming language syntax relates to the structure, format, combination of symbols/numbers/letters and even spelling

Don't worry about making mistakes.



Let's get coding...

