How to use the *Pythonista* app to learn the *Python* programming language
Press ☑️ and create an Empty Script. Type in these commands and run them ▶️:

```python
print 'Hello world.'
print 'I can print text.'
print '\n'
print 'Why did the beach cry?'
print 'Because the seaweed!
```

**Key vocabulary**

**Program** – A sequence of commands which a computer follows.

**Run** - Carrying out the commands in a program.
2. **Solving calculations**

Press ☑ and create an Empty Script. Type in these commands and run them ▶:

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>print 50 + 50</td>
</tr>
<tr>
<td>print 50 − 25</td>
</tr>
<tr>
<td>print 50 * 10</td>
</tr>
<tr>
<td>print 50 / 5</td>
</tr>
<tr>
<td>print (3 * 6) + 2</td>
</tr>
<tr>
<td>print (8 + 7) / 3</td>
</tr>
<tr>
<td>print (20 − 10) * 5</td>
</tr>
</tbody>
</table>

The answers to these calculations will be printed when the program is run.

- Can you change the numbers in the calculations? Remember to press ▶ to check your program works!

**Key vocabulary**

- **Testing** - Trying out a program to check if it works as expected.
- **Debugging** - Finding and correcting mistakes in a program's code.
3. **Text variables**

Press 🖋 and create an **Empty Script**. Type in these commands and run them:

```python
name = 'Molly'
print 'Hello', name
food = 'chocolate'
print 'I like to eat', food
team = 'Manchester United'
print 'I support', team

sport = swimming
print sport, 'is fun'
```

- **Key vocabulary**
  - **Variable** – A value that can be stored and used in a program.

---

This is a variable – something that can change.

You always put text in inverted commas!

This prints what the variable is set as.

---

**Edit and improve:**

- Change the text each variable is set as. Remember the inverted commas!
4. **Inputting text**

Press and create an Empty Script. Type in these commands and run them:

This is a variable – something that can change.

The `raw_input` command makes the user type in what they want.

```python
name = raw_input('What is your name?')
print 'Hello', name

age = raw_input('What is your age?')
print 'You are', age

town = raw_input('Where do you live?')
print 'You live in', town

subject = raw_input('What subject do you like?')
print subject, 'is your favourite'
```
5. **Inputting numbers**

Press *and create an Empty Script*. Type in these commands and run them *:

```python
number = float(input('Type a whole number.'))
answer = number * 8
print answer

number2 = float(input('Type another whole number.'))
answer = number + number 2
print answer
```

- Change this symbol to do different calculations.

This is a variable—something that can change.

The `float` command tells the computer the user is typing a number.
6. **Random numbers**

Press 📝 and create an Empty Script. Type in these commands and run them:

```python
import random

number = random.randrange(10, 20, 1)
print number
print number + 10
print number * 10
```

- This sets a variable as a random number.
- What happens if you change the 10 to a smaller number or the 20 to a bigger number?

**Edit and improve:**

- Change these symbols and numbers to do different calculations with the random number.
Press and create an Empty Script. Type in these commands and run them:

```python
score = float(input('Type your score.'))
total = float(input('Type the total possible.'))
percent = score / total * 100
print 'Your percentage is', percent

width = float(input('Type the rectangle width.'))
length = float(input('Type the rectangle length.'))
area = length * width
print 'The area is', area
```

This works out what % you got in a test.

This works out the area of a rectangle.

Programming challenge:

Create a program that calculates the perimeter of a rectangle by adding together its two lengths and two widths, inputted by the user.
8. Lists

Press \[\text{ wealthy } \] and create an \textit{Empty Script}. Type in these commands and run them ▶️:

```python
import random

colours = ['red', 'green']
animals = ['lions', 'bears']

print 'My rainbow zoo has:'

colour = random.choice(colours)
animal = random.choice(animals)
print colour, animal
```

- Put more items in the lists to make the rainbow zoo more fun!

**Key vocabulary**

- **List** – A set of values

---

These two variables store lists of colours and animals. Remember your inverted commas!

This picks a random colour and animal and prints it.
Press ☀️ and create an Empty Script. Type in these commands and run them:

```python
import random
def cointoss():
    options = ['heads', 'tails']
    result = random.choice(options)
    print(result)
cointoss()
cointoss()
cointoss()
cointoss()
cointoss()
```

**Key vocabulary**

**Function** – A sub-program which can be called (run) later using its name.

This function is named cointoss.

This is a function – a set of commands with a name that does something (tosses a coin).

**Programming challenge:**

Change this function to roll a dice instead. Change its name from `cointoss` to `roll`.
Then change the options to `[1, 2, 3, 4, 5, 6]`
Press and create an Empty Script. Type in these commands and run them:

```python
answer = raw_input('Do cats bark? ')  # This is the question being asked
if answer == 'no':  # This is the condition being checked
    print 'Correct'
else:
    print 'Wrong'
```

This program prints ‘Correct’ if the user answers correctly, else prints ‘Wrong’ if they answer incorrectly.

**REMEmBR THE COLON!**

**THIS MUST BE TABBED IN!**

== means ‘is the same as’

**Edit and improve:**

- Change the question being asked (and the answer too, if needed).
11. **OR statements**

Press and create an Empty Script. Type in these commands and run them:

```python
answer = raw_input('Is it dark at night? ')
if answer == 'yes' or answer == 'YES':
    print 'Correct'
else:
    print 'Wrong'
```

The **or** command lets the user input different answers but still get the question correct.

**Key vocabulary**

**Conditional (IF) statement** – Decides which commands to run depending on whether certain things (conditions) are true or false.

- Change the question being asked (and the answer too, if needed).
Press and create an Empty Script. Type in these commands and run them:

```python
score = 0

answer = raw_input('Is it grass green? ')
if answer == 'yes' or answer == 'YES':
    print 'Correct'
    score = score + 1
else:
    print 'Wrong'

answer = raw_input('What is 3 + 3? ')
if answer == '6' or answer == 'six':
    print 'Correct'
    score = score + 1
else:
    print 'Wrong'

print 'Your score is', score
```

This variable sets the score to 0 at the start.

This adds 1 to the score if the user answers correctly.

The quiz uses conditional (if) statements to print if the user answers correctly or not.

**Programming challenge:**
Create your own quiz that prints the player’s score at the end.
while loops

Press  and create an Empty Script. Type in these commands and run them:

```python
password = 'fish'
guess = ''

while (password != guess):
    guess = raw_input('Enter password: ')
    if password == guess:
        print 'Correct'
    else:
        print 'Try again'
```

**Key vocabulary**

**While loop** – Commands in a while loop keep repeating until a condition is met (e.g. the correct password is inputted).