# Discussion Unit 1

1. >>> print 'Hello, World!'
   1. >>> print 'Hello, World!'
   2. File "<stdin>", line 1
   3. print 'Hello, World!'
   4. ^^^^^^^^^^^^^^^^^^^^^
   5. SyntaxError: Missing parentheses in call to 'print'. Did you mean print(...)?

**Explanation**:

* This is a print statement for python 2 and not python 3. There is a slight difference between python 2 and python 3.
* For this to run effectively in python 3, we need a parentheses that indicates print function, quotation marks to indicate the beginning and the end of a text (result) to be displayed. For example >>> print ('Hello, World!') will display Hello, World!

1. >>> 1/2
2. >>> 1/2

0.5

**Explanation**:

* This is an arithmetic operator (/)
* An integer is divided by another integer
* The results is of type <class 'float'> 0.5
* A similar example can be >>> 84/2 resulting to 42.0. In Python 3, division with the "/" operator always returns a floating-point number, even if the result is a whole number. This behavior ensures that the result is accurate regardless of whether the division produces a fractional part.

1. >>> type(1/2)
2. >>> type (1/2)

<class 'float'>

**Explanation**:

* This is a syntax used in python 3 when trying to get the value type of a statement.
* The result this without the type will be >>> 1/2 which will be 0.5. This belongs to the type class float. The reason our given output is <class 'float'>

1. >>> print(01)
2. print (01)
3. File "<stdin>", line 1
4. print (01)
5. ^

SyntaxError: leading zeros in decimal integer literals are not permitted; use an 0o prefix for octal integers

**Explanation**:

* This output indicates and error. Leading zeros in integer literals indicate an octal (base 8) number. However, in this case, since 01 is less than 8, it's interpreted as a decimal number
* If we >>> print (0) or >>> print (1), it will output 0 and 1 respectively. This is because I am just telling the interpreter to display the value 0 or 1 to the console
* >>> print ('01')
* 01
* >>> print (0)
* 0
* >>> print (1)

1

* For this to be printed without any errors, we have to print as a string >>> print ('01') which will output 01 as seen above.

1. >>> print 1/(2/3)
2. >>> 1/(2/3)

1.5

**Explanation**:

* We are evaluating the expression inside the parentheses 2/3.
* >>> 2/3 evaluates to approximately 0.6666666666666666 , because in Python, division of two integers results in a floating-point number.
* Now, we divide 1 by the result of 2/3, which is 0.6666666666666666.
* So, 1 divided by 0.6666666666666666 is approximately 1.5 as outputted above.