

Learning Goals/Objectives

Be able to read, comprehend, trace, adapt and create Python code that uses:

- Data types & casting - understanding the different data formats that variables & lists use, and how to convert between them.
- Random - how to generate random numbers and use them in programs
- Modulo - how to calculate the remainder of an integer division and why this is useful
-

Data Types

What is a data type?

- A **data type** is a setting for a **variable**. It tells the variable what sort of data it will store.
- At the moment, we have only used two data types, the **string** and the **int**.
- What type of data does a **string** store?
- What type of data does an **int** store?

Data Types in Python

String - Text

Int - Whole numbers

Float (also called *Real* in other coding languages) - numbers with decimals.

Type - How To Code

The **type()** function returns the data type of some data or a variable

Put the data in the brackets as a parameter.

```
type(5.4)
```

OR put the variable containing the data in the brackets as a parameter.

```
num1 = 5
```

```
print(type(num1))
```

Programming – Casting (Changing One Data Type To Another)

```
int(data/variable)  
float(data/variable)  
str(data/variable)
```

Programming – Casting Int

```
x = int(1)           # x will be 1  
y = int(2.8)         # y will be 2  
z = int("3")         # z will be 3
```

Programming – Casting Float

```
x = float(1)
```

```
# x will be 1.0
```

```
y = float(2.8)
```

```
# y will be 2.8
```

```
z = float("3")
```

```
# z will be 3.0
```

```
w = float("4.2")
```

```
# w will be 4.2
```

Programming – Casting String

```
x = str("s1")
```

```
y = str(2)
```

```
z = str(3.0)
```

```
# x will be 's1'
```

```
# y will be '2'
```

```
# z will be '3.0'
```

Casting - How To Code

Casting lets us convert one data type into another.

1. Put the name of the data type you want to convert to.

2. Put the data OR the variable in brackets as the parameter.

```
int(data/variable)  
float(data/variable)  
str(data/variable)
```

Task - Predict & Run

```
1  # Task
2
3  # Add comments to the code to predict what the code does and what the output will be.
4
5  data1 = 2.9
6  data2 = "Hello World!"
7  data3 = 6
8
9  print(type(data1))
10 print(type(data3))
11 print(type(data2))
12
13 data1 = data1 + 0.1
14
15 print(type(data1))
16
17 data3 = float(data3)
18 data1 = int(data1)
19
20 print(type(data1))
21 print(type(data3))
--
```

Task - Investigate & Modify

```
1  # Task - Investigate
2
3  num1 = int(input("Enter a number"))
4  num2 = float(input("Enter another number"))
5
6  total = num1 * num2
7
8  print(total)
9
10  # What does the 'int' before 'input' make the program do?
11
12  # What does the 'float' before 'input' make the program do?
13
14  # What data type will the output be?
15
16
17  #Task - Modify
18
19  #Adapt the code to:
20
21  # Get both inputs as floats
22  # Convert the total to an int before it is output.
23
```

<https://repl.it/@MrAColley/2202-Type-and-Cast-Investigate-and-Modify>



Task - Make

```
1  # Task - Make
2
3  #Write a program to calculate the area of a rectangle.
4
5  # Get input for height & width as floats.
6
7  #Multiply the height & width to calculate the area.
8
9  #Output the area as part of a sentence.
10
11 #Extra challenge - create this as a function with a suitable name
    that takes fixed height & width as its parameters.
12
13 # Extra extra challenge - create this as a function with a suitable
    name that takes user input as its parameters.
14
```

<https://repl.it/@MrAColley/2203-Type-and-Cast-Make>



Random Numbers

Random Numbers in Python

Python has lots of pre-written features & functions that we can use. Often, these features are grouped together and called **libraries**. To use a library we have to **import** it (you only have to do this once). It is common practice to put all of your imports at the **top** of your code.

The library we are going to use for random numbers is called **random**

Random- How To Code

The **randint(x,y)** function takes 2 parameters.

x is the lowest random number that can be picked.

y is the largest.

The program will pick a number between these two limits.

1. Import the library
(you only have to
do this **once**)

```
import random
```

2. Type the name of
the library, a . and
the name of the
function you want to
use.

```
random.randint(1,20)
```

3. Put your
upper and
lower limits in
the brackets
as
parameters.

Task - Predict & Run

```
1  # Task Predict & Run
2
3  #Add comments to explain what the code does and what the output will be.
4
5  import random
6
7  print(random.randint(1,5))
8
9  num1 = random.randint(1,10)
10
11 print("Your number was " + str(num1))
12
```

Task - Investigate & Modify

```
1  # Task - Investigate
2
3  # Answer the questions below the code.
4  import random
5
6  print("Welcome to the dice simulator!")
7
8  num1 = random.randint(1,6)
9
10 print("You rolled a " + str(num1))
11
12 # What is the term for the (1,6) values used by the randint function?
13
14 # Why are the numbers in brackets not (0,6)
15
16 # What would the effect be if the last two lines of code swapped places?
17
18 # Task - Modify
19
20 #Adapt the code so that
21
22 #It generates a second number by rolling the dice again.
23
24 #It adds the two dice rolls together.
25
26 #It outputs the total of the two dice rolls
```



Task - Make

```
1  # Task - Make
2
3  #Write a program that:
4
5  #Gets user input of two numbers.
6
7  # Extra challenge - Build in a check for the input, if the second number is lower than or
  the same as the first number then output an error message. Else continue to the next steps.
8
9  #Generates a random number between the two numbers input.
10
11 # Outputs the random number generated.
12
```



<https://repl.it/@MrAColley/2206-Random-Make>

Modulus

Modulus in Python

3 MOD 1 = 3 remainder 0, so the value returned would be 0.

5 MOD 2 = 2 remainder 1, so the value returned would be 1.

14 MOD 4 = 3 remainder 2, so the value returned would be 2.

What Value Will Be Returned By....

7 MOD 6

29 MOD 4

15 MOD 5

9 MOD 7

35 MOD 4

Modulus - How To Code

The **modulus** operator returns the **remainder** of integer division.

1. Put the dividend on the left.

2. Use the % symbol for modulus.

3. Put the divisor on the right.

12 % 5

Modulus With Variables

The **modulus** operator returns the **remainder** of integer division.

```
num1 = 12  
num2 = 5  
remainder = num1 % num2
```

Task - Predict & Run

```
1  # Task - Predict & Run
2
3  # Add comments to the code to explain what it does and what the output
   # will be.
4
5  # Run the code to test your predictions.
6
7  8 % 3
8  9 % 3
9  14 % 5
10
11 remainder = 10 % 4
12 print(remainder)
13
14 num1 = 16
15 num2 = 4
16
17 remainder = num1 % num2
18 print("The remainder is " + remainder)
19
```

Task - Investigate

```
# Task - Investigate

# Answer the questions about the code below. Type your answers as comments.

num1 = int(input("Enter a number"))

print ("I will now calculate if your number is in the two times table.")

remainder = num1 % 2

if remainder == 0:
    print("Your number is in the two times table")
else:
    print("Your number is not in the two times table")

# What is the purpose of the code?

# What symbol is used for modulus?

# What would happen if the input to the program was 17?

# What would happen if the input to the program was 98?

# What is the condition in the code?

# Why can the code use 'else' instead of 'elif' and still work correctly?
```



<https://repl.it/@MrAColley/2208-Modulus-Investigate-and-Modify>

Resources created by Andy Colley (@MrAColley)

Task - Modify

```
# Task - Modify
```

```
# Complete the code below so that:
```

```
# It gets user input into the 'name variable'
```

```
# It uses selection to output suitable messages depending on whether  
the name has an odd or even number of characters
```

```
# Remove the need for the 'nameLength' and 'nameRemainder' variables by  
using len and modulus in the condition for the selection.
```

```
name = "Dave"
```

```
nameLength = len(name)
```

```
nameRemainder = nameLength % 2
```

```
if
```

Task - Make

```
1  # Task - Make - The Love Calculator
2
3  # Write a program that:
4
5  # Gets two users to input their names.
6  # Calulates the number of characters in each name and adds them together.
7  # Calculates the modulus of the total characters in both names divided by
   3.
8  # If the modulus is 0, output a message saying that the couple are very
   compatible.
9  # If the modulus is 1, output a message saying that the couple are might
   have a chance together.
10 # If the modulus is 2, output a message saying that the couple aren't
    compatible.
```