techtalent academy

Starter Activity.

Which is the **easiest** to program to say "**Hello World**"?

1. Assembler

```
org 100h
main  proc
      mov    ah,9              ;
      mov    dx,offset hello_message
      int    21h               :
retn
hello_message db 'Hello, world!$'
main  endp
end   main
```

2. C++

```
#include <iostream>
int main()
{
  std::cout << “Hello World!” << std::endl;
  return 0;
}
```

3. Python

```
print (“Hello, World!”)
```

techtalent academy

TechTalent Academy

Data Science Course

Python Fundamentals Part 1

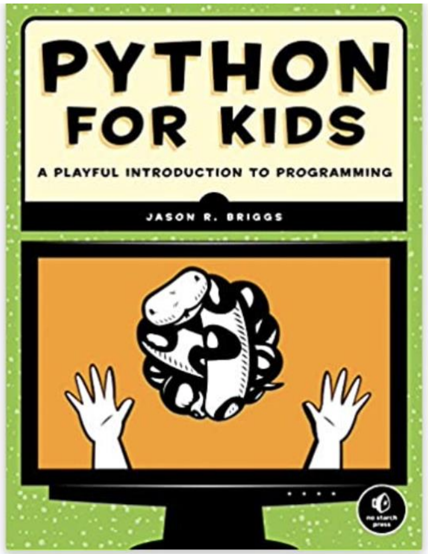
Lesson Objectives.

- How to declare variables and Python datatypes
- Basic Math expressions
- How to assign types to data (data casting)

Books and Links.

techtalent academy

For complete beginners

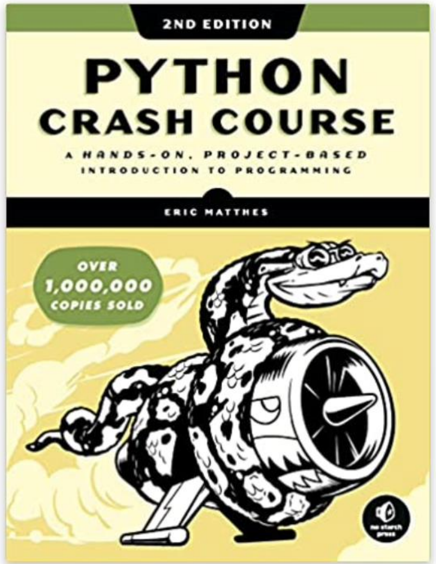


Python for Kids: A Playful Introduction to Programming, by Jason R. Briggs

w3schools

https://www.w3schools.com/python/

For intermediate/advanced


2ND EDITION
PYTHON CRASH COURSE
A HANDS-ON, PROJECT-BASED INTRODUCTION TO PROGRAMMING
ERIC MATTHES
OVER 1,000,000 COPIES SOLD

Python Crash Course, 2nd Edition: A Hands-On, Project-Based Introduction To Programming, by Eric Matthes

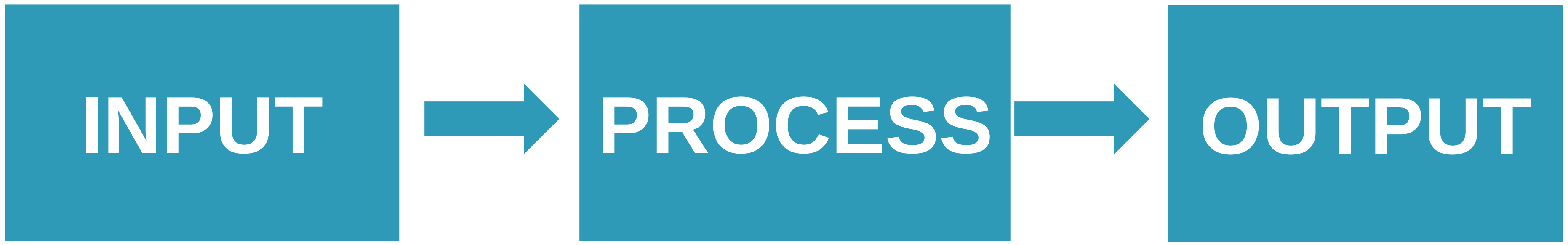
Real Python

https://realpython.com

What is a Computer System?

A computer is a system that takes a set of digital inputs, processes them, and creates an output.

This is done using a combination of hardware and software



What is a Program?

Whenever we use a computer, phone or tablet to undertake any activity, this has been facilitated using programming. This includes operating systems (Windows, Mac OS, Linux), browsers and any apps that you use, to name just a few.

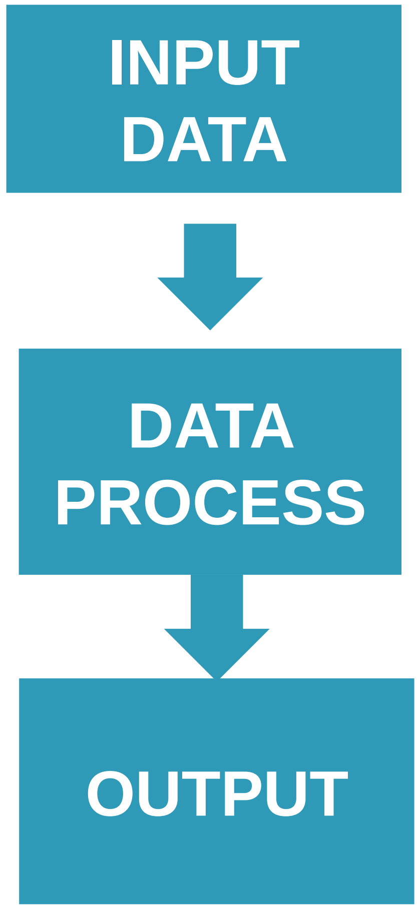
As users, you may often interact with a front-end user interface (UI). “Underneath” this user interface, lines of code are being run to enable the computation to perform the necessary operations.

The sequence of **instructions** that specifies how to perform the computation is known as a **program**. These sets of instructions are written by **programmers**.

Programming is not telling the computer what you want to do. It is telling it **EXACTLY** how to do it.

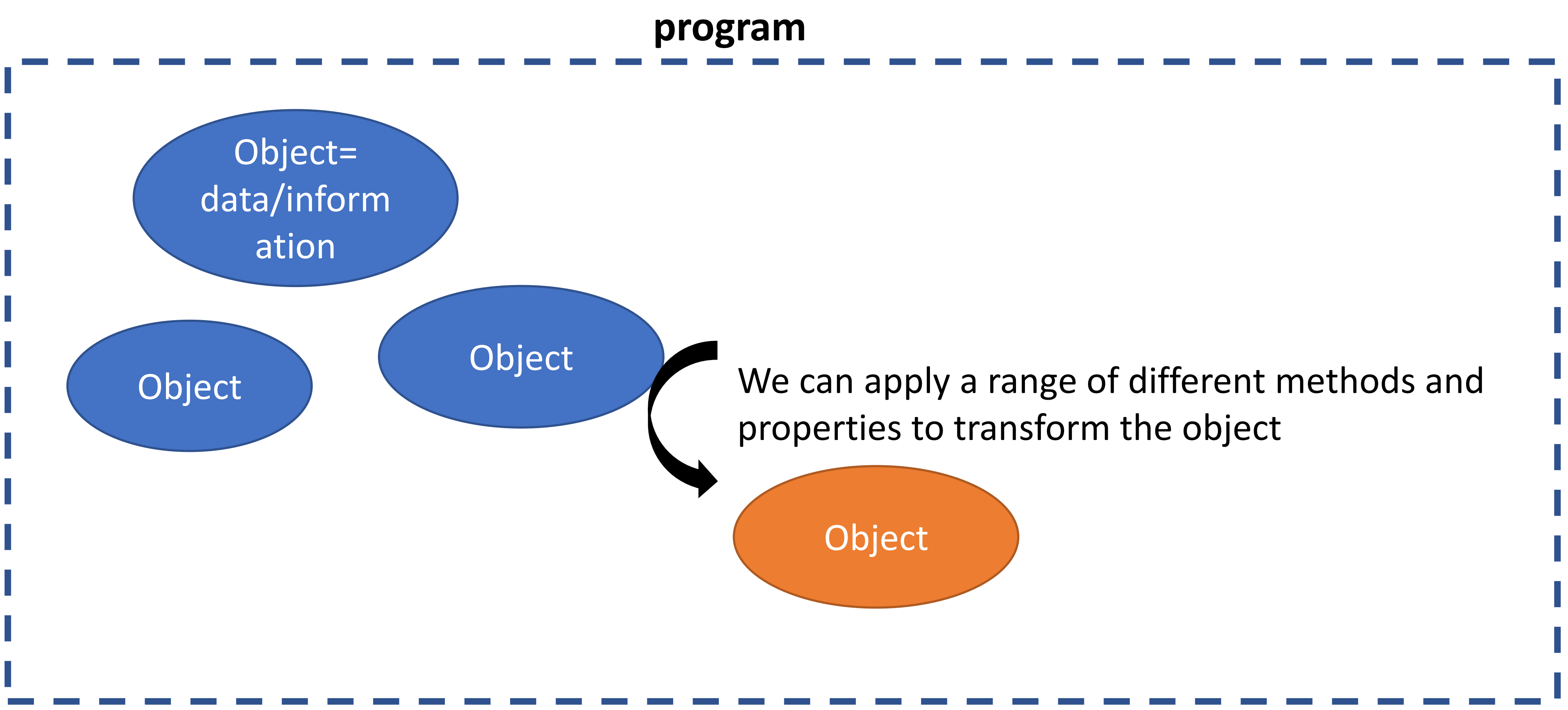
What is a Program.

Your Python program will have a set of instructions read from top to bottom



Instruction # 1
Instruction # 2
Instruction # 3

Python is an Object-Oriented Programming language.



We can apply a range of different methods and properties to transform the object

In Python objects are Class based

techtalent academy

History of Python.

- Python is a widely used, programming language. It was originally designed by Guido van Rossum and released in the early 1990's
- It came off the back of the ABC language
- The name was inspired by "Monty Python"
- Known for its simplistic, concise and modular approach, it gained momentum from the start and is now one of the most popular programming languages
- Particularly useful for data analysis
- Currently on Python 3.10

Why is Python used In Data Science?

Python provide great functionality to deal with mathematics, statistics and scientific function. It provides great libraries to deals with data science application

Numpy: Python library that provides mathematical function to handle large dimension array. It provides various method/function for Array, Metrics, and linear algebra.

Pandas: Pandas is one of the most popular Python library for data manipulation and analysis. Pandas provide useful functions to manipulate large amount of structured data. Pandas provide easiest method to perform analysis.

Matplotlib: Matplotlib is another useful Python library for Data Visualization. Descriptive analysis and visualizing data is very important for any organization. Matplotlib provides various method to Visualize data in more effective way.

Scikit – learn: Sklearn is Python library for machine learning. Sklearn provides various algorithms and functions that are used in machine learning. Sklearn provides easy and simple tools for data mining and data analysis.

Practice: Using VS Code.

- <u>Open a new file:</u>
  - We need to save the file first. If the file isn't saved, the computer won't know what to run!
    Save the file as **python-practice.ipynb**
  - We have to save the file and then point the computer in the right direction…

- In the script type:
  - **print("Hello World")**

- <u>Run the file selecting Run or holding crtl and enter at the same time:</u>

  •Output should be: **Hello World**

  •What happens if you run **print('Hello World')or print(Hello World)**?

The print() function is used to execute code

Syntax.

The **syntax** of a **programming language** is the set of rules that apply when writing in that language

Syntax is made up of special key words, characters, and symbols which are combined to result in a program

It is important to remember that **ALL** syntax needs to be correct for the program to run

Syntax Errors.

- Task: Correct the errors
- Your programs won't always work first time!

- Make sure to check:
  - Spelling – **pront("Hello World")** returns a **NameError**
  - Punctuation – Python3 requires brackets at the print line:
    print "Hello World" returns a **SyntaxError**
  - Logic – Syntax is correct, but program produces an incorrect result. In this case, more testing is required.

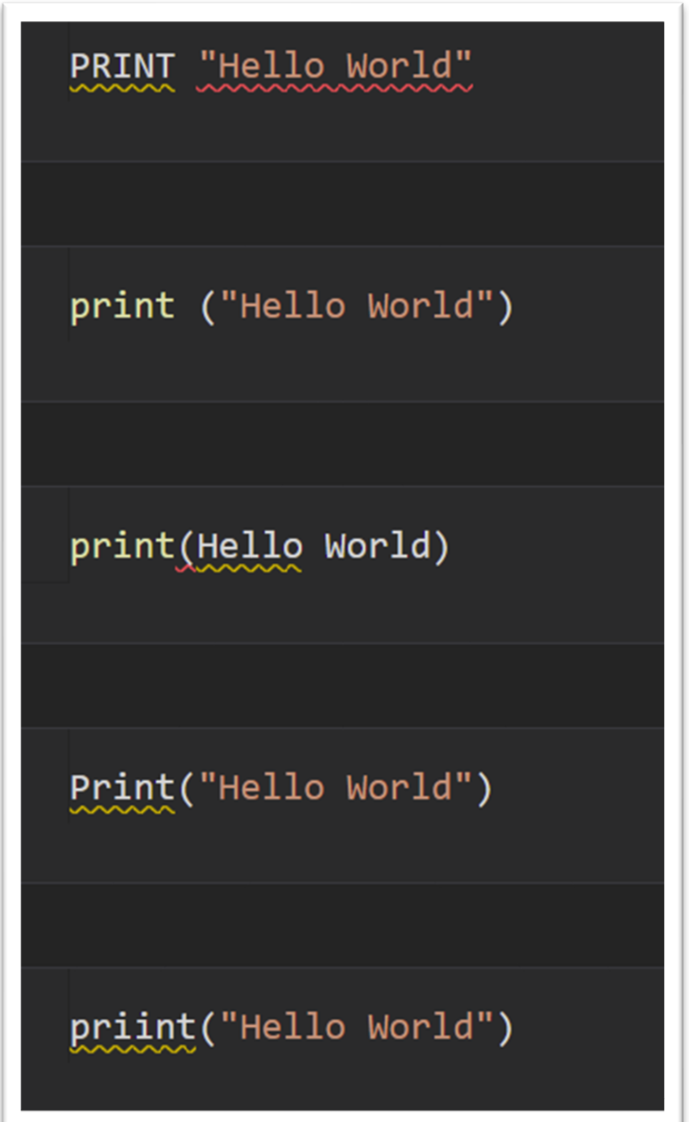
- IDEs can usually help spot errors using the colour of the text, but it is important not to become over-reliant on this!

- You must debug, spot the error, correct and re-run

Task: Correct the errors.

- Will these run?
- Spot the error, correct and re-run.

- Code will not always work first time and that is OKAY!
- Learning how to debug code is a very important skill.



```
PRINT "Hello World"

print ("Hello World")

print(Hello World)

Print("Hello World")

priint("Hello World")
```

Comments.

techtalent academy

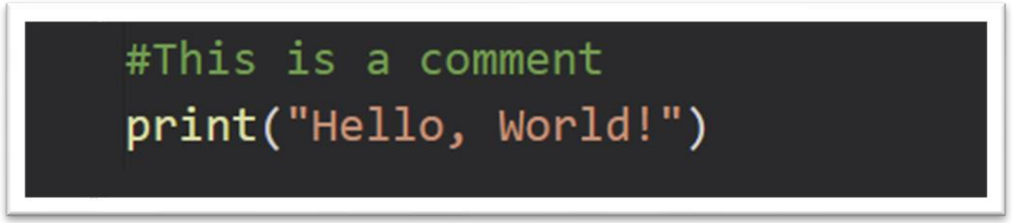
Commenting is important in coding

It describes what is going on

It acts as a reminder, when you go back and look at your code

The idea is for someone who has never seen the code to be able to read it, understand it and change it

To create a comment within your code, you need to use the # symbol. Your program will ignore the comments and will run everything else



```python
#This is a comment
print("Hello, World!")
```

Variables.

In many cases, we need to be able to store data within our program. One way to do this is to use a **variable**.

The best way to think of a variable is as a container for a piece of data.

Syntax:

```
variable_name = data
```

The name you want to give your variable.

Using the equals sign between your variable name and your variable data tells you program that you want **to assign the data to the variable.**

The data value you wish to be contained within the variable.

Variables.

Think variables as storage boxes or containers that contain particular items.

Name you variables based on what they contained.

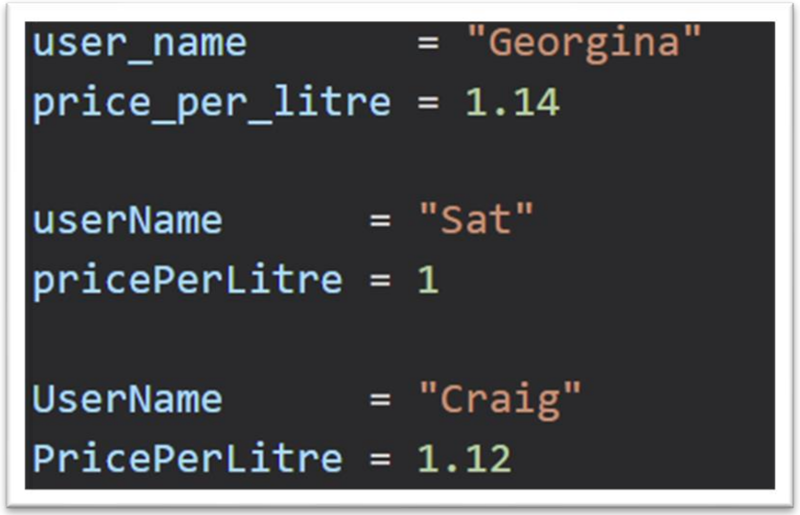
book_box

music_box

Variables.

Variables must have an identifiable name. It is best practice to ensure that your naming convention is consistent.

Remember that Python is also **case sensitive**!
Variable name must start ideally with a character (a-z), you can use an underscore sign to separate the words



```
user_name       = "Georgina"
price_per_litre = 1.14

userName      = "Sat"
pricePerLitre = 1

UserName      = "Craig"
PricePerLitre = 1.12
```

Try on VS Code!

Task: Variables.

1. Create a variable that will contain your first name (name your variable accordingly)

2. Create a variable that will contain your favourite number (name your variable accordingly)

Data Types.

Data types tell your interpreter (and therefore your program) how you intend to use the data.

In a program you may wish perform certain operations on your data, such as:

- conducting a calculation
- displaying a series of words to a user
- setting up a data structure

In order to ensure that our programs can perform these operations without issue, it is essential that we assign the correct data type to our data.

Data Type	Description	Example
String	Strings hold a list of characters, must use single or double quotations when entered	“iPhone 12” or ‘iPhone12’ Or ‘1’
Integer	A whole number without a decimal point can be positive or negative	0 12
Float	A number which includes a decimal point can be positive or negative	2.5 5.355
Boolean	A Boolean variable can only have one of two values, TRUE or FALSE	True False

2 is NOT the same as **2.0** which is NOT the same as **“2”**

More complex data types (next lesson in details!)

Data Type	Description	Example
List []	A list object is an ordered collection of one or more items of data, this does need to be the same type, but must be put in square brackets. Can be modified.	[“1”,2,3,4]
Tuple ()	A Tuple object is an ordered collection of one or more items of data, this does not need to be of the same type but must be put inside parentheses. Cannot be changed.	(1,2,3,4)
Dictionary {}	A dictionary object is an unordered collection of data in a {**key: value, key2:value2} pair form**. A collection of such pairs is enclosed in curly brackets. Can be modified	{1:”one”,2:”two”,3:”three”}

Task: Datatypes.

Create 4 different variables which store data in the following datatypes:

- A string
- An integer number
- A float number
- A boolean

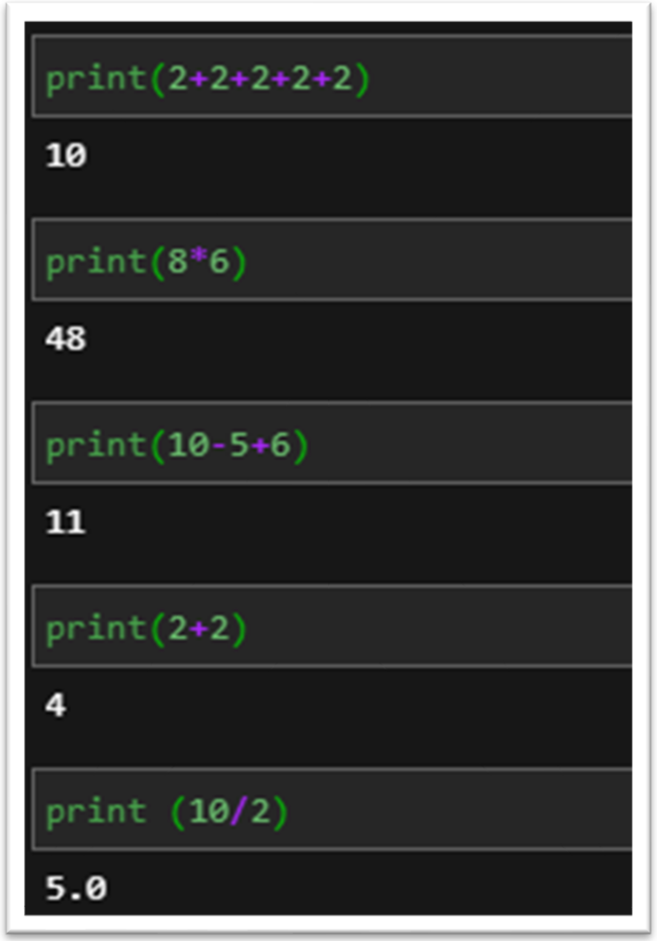
techtalent academy

Mathematical Expressions.

Python can perform mathematical computations easily.

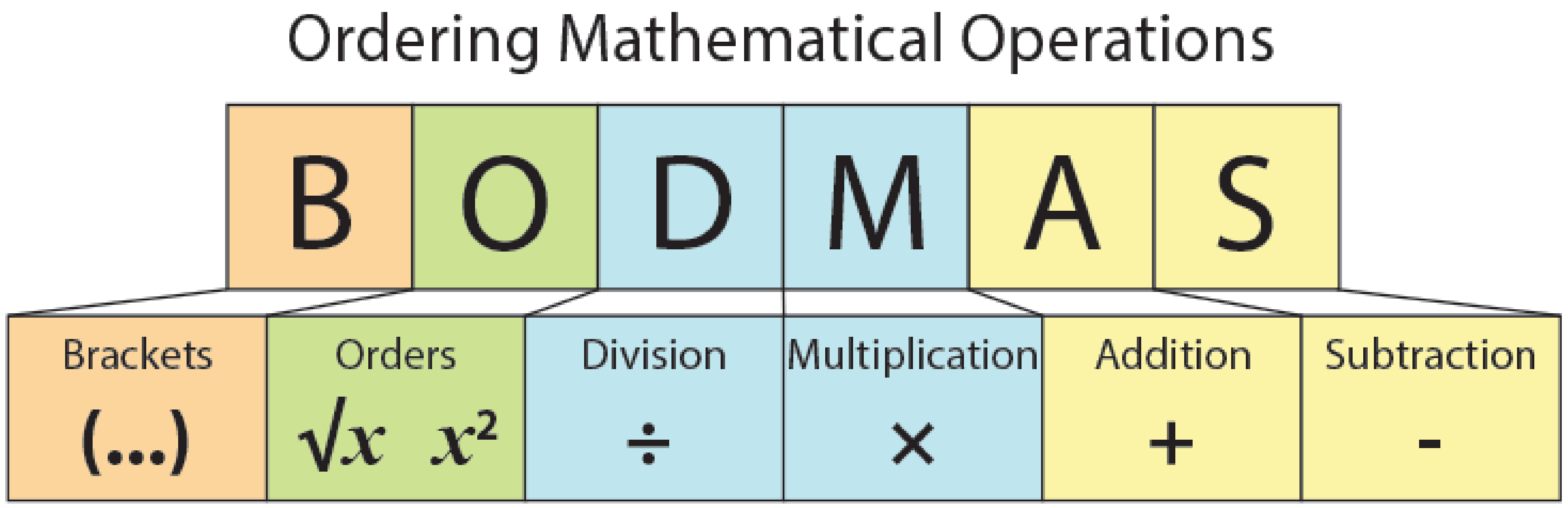
Basic mathematics can be performed on Python:

Addition: +
Subtraction: -
Multiplication: *
Division : /
Powers : **
Modulus: %



```
print(2+2+2+2+2)
10
print(8*6)
48
print(10-5+6)
11
print(2+2)
4
print (10/2)
5.0
```

BODMAS.



- This is the order in which mathematical operations are executed.
- This is also how they are ordered in Python!

[Example: https://www.youtube.com/watch?v=70cAYYCJBuQ](https://www.youtube.com/watch?v=70cAYYCJBuQ)

Python built-in function.

- Python come with a variety of functions that can be used directly to perform some actions on your data (data are usually stored in variables)

```
syntax:

function_name()
```

```
syntax:

function_name(data)
```

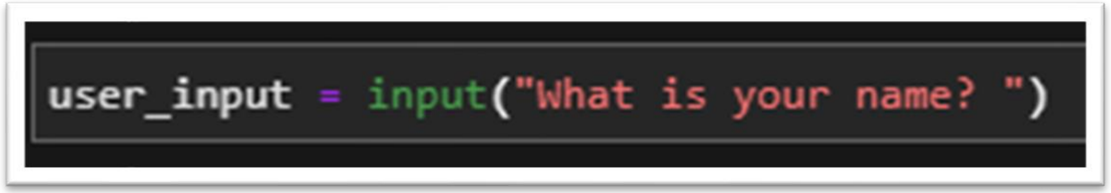
Data/variable can be added inside the function parenthesis

https://www.w3schools.com/python/python_ref_functions.asp

Python built-in function: User Input.

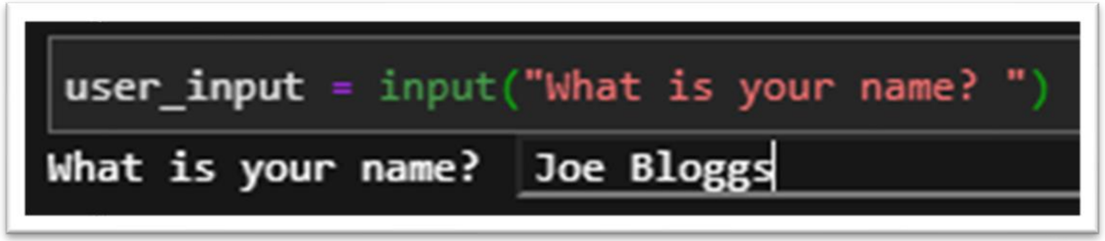
User input allow you to ask a user for an input, and then store that input in a variable.

To set up a variable as requiring an input, you need to specify that you wish to record a user input.



```
user_input = input("What is your name? ")
```

When you run this, you will note that there is a text box that appears, enabling the user to type and store their details.



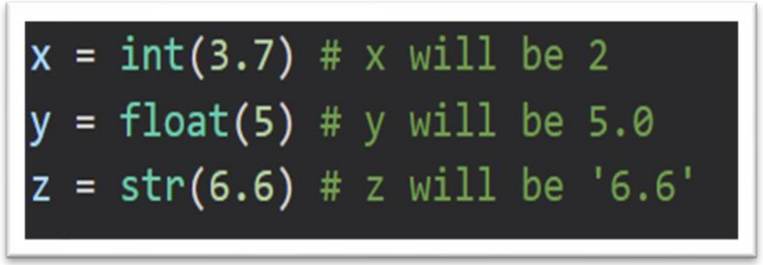
```
user_input = input("What is your name? ")
What is your name? Joe Bloggs
```

In the above case, the variable **user_input** will hold the data **Joe Bloggs**, based on the data entered by the user.

techtalent academy

Python built-in function for Casting.

Casting is the conversion of one Data Type to another. So converting an Integer into a String or vice versa!

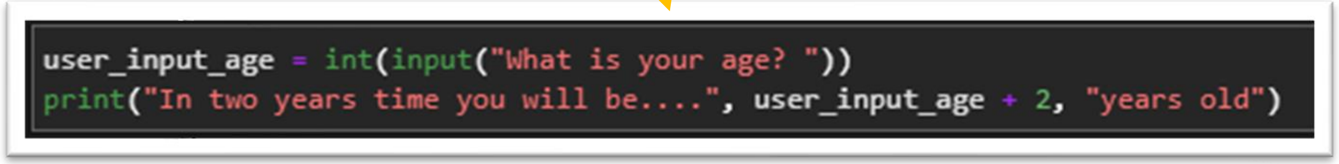


```
x = int(3.7) # x will be 2
y = float(5) # y will be 5.0
z = str(6.6) # z will be '6.6'
```

To cast data as a specific data type you must specify the type outside of parentheses containing the data.

int = integer
float = float
str = string

When asking for **user input**, the default data type will be **string**. For example, asking a user their age, the response will be stored as a string unless you specify that you want to record this as an integer.



```
user_input_age = int(input("What is your age? "))
print("In two years time you will be....", user_input_age + 2, "years old")
```

Task: User Input.

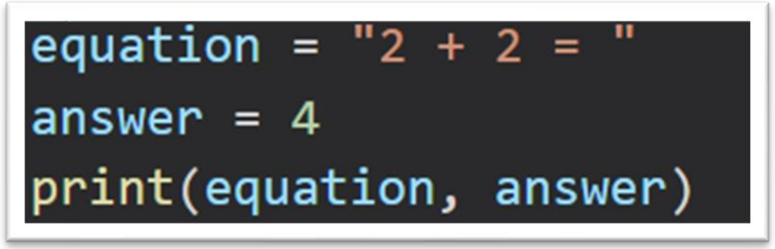
- Ask for the user's first name and then ask for their surname and display the output message:

  Hello! My name is First Name and Surname

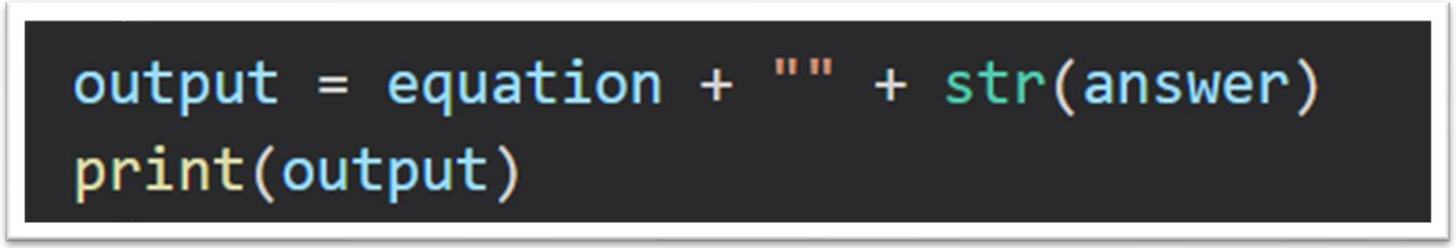
- Ask the user to enter a number over 100 and then enter a number under 10 and tell them how many times the smaller number goes into the larger number in a user-friendly format.

String concatenation & Line Breaks.

- **Concatenation:**
  -Combine strings using a: **+**
  -Combine text and numbers using a comma: **,**
  -For example:

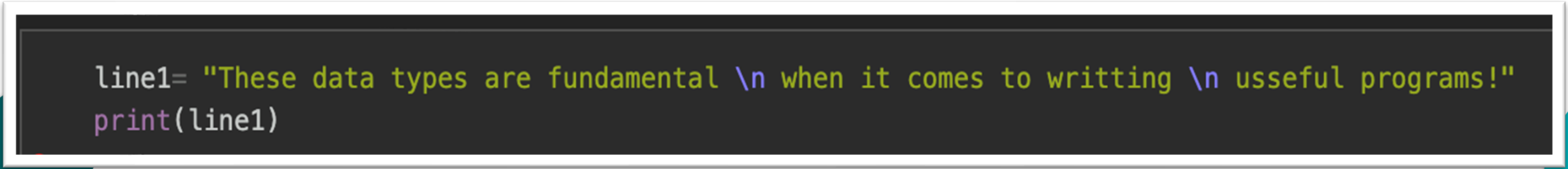


```
equation = "2 + 2 = "
answer = 4
print(equation, answer)
```



```
output = equation + "" + str(answer)
print(output)
```

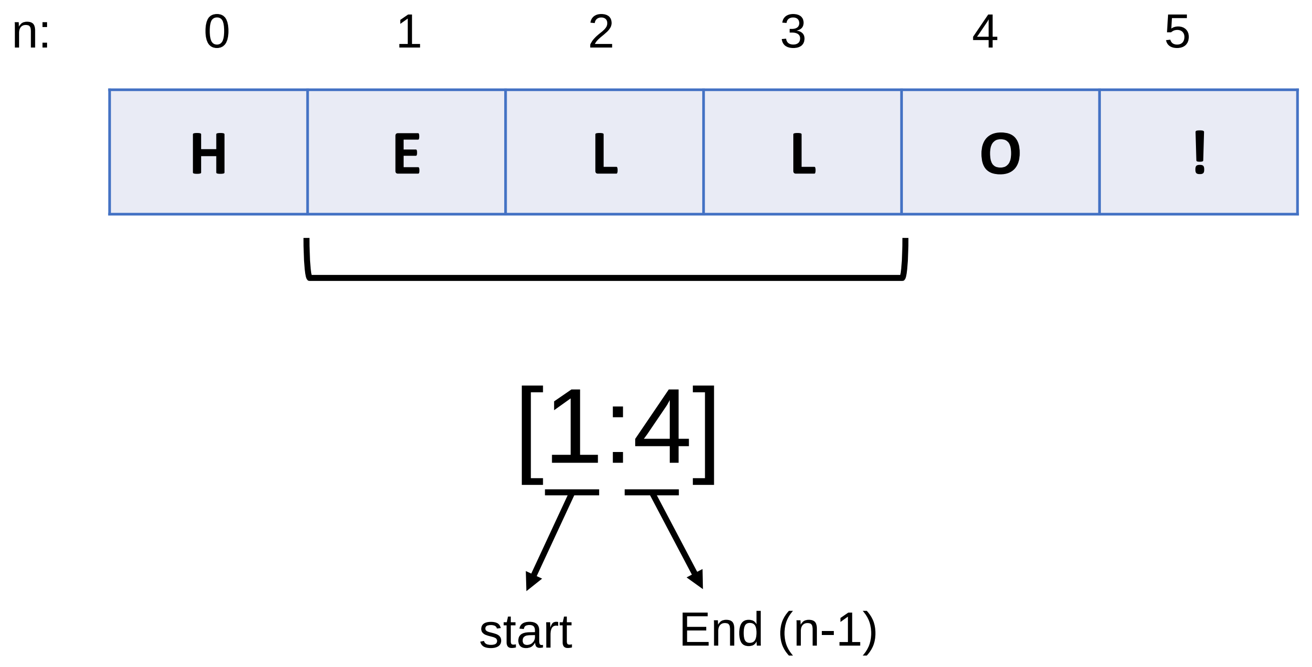
-Strings can be written over several lines using a **line break**: **\n**



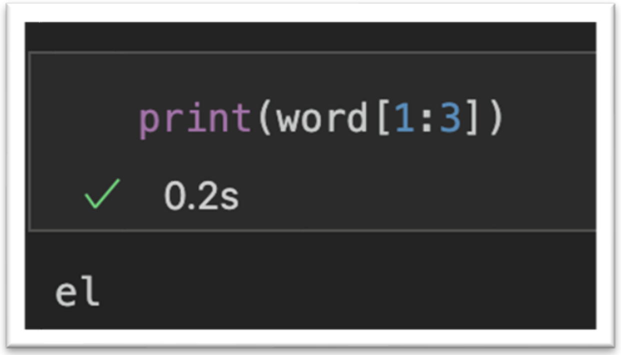
```
line1= "These data types are fundamental \n when it comes to writting \n usseful programs!"
print(line1)
```

Slicing Strings.

- You can return a range of characters by using the slice syntax.
- Specify the start and end index , which is separated by a colon symbol : in a square bracket []

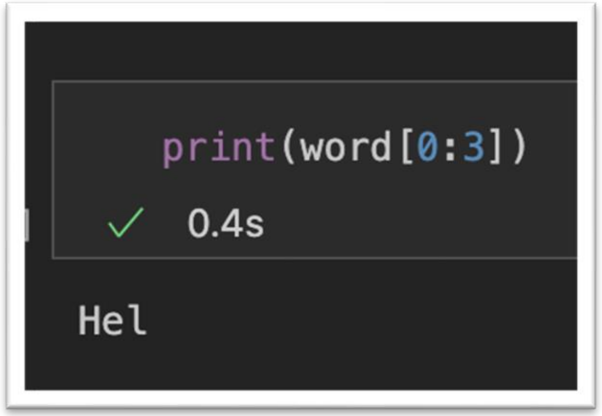


Slicing Strings.



```
print(word[1:3])
✓ 0.2s
el
```

Returns characters from position 1 to position 3 (not included)

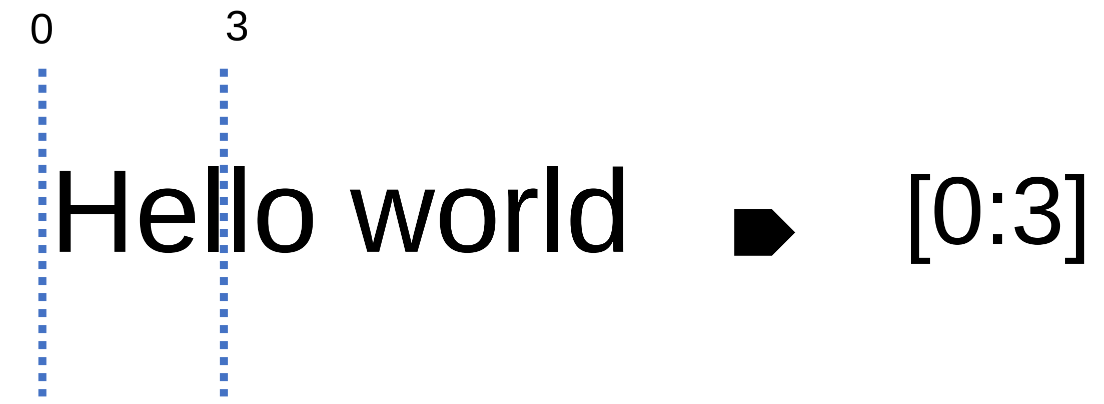


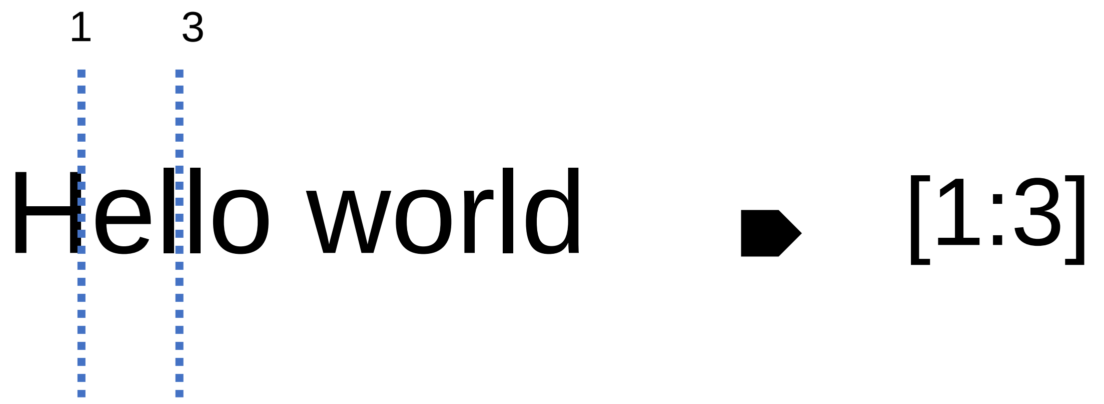
```
print(word[0:3])
✓ 0.4s
Hel
```

Returns characters from position 0 to position 3 (not included)

Slicing Strings.

Alternative method:



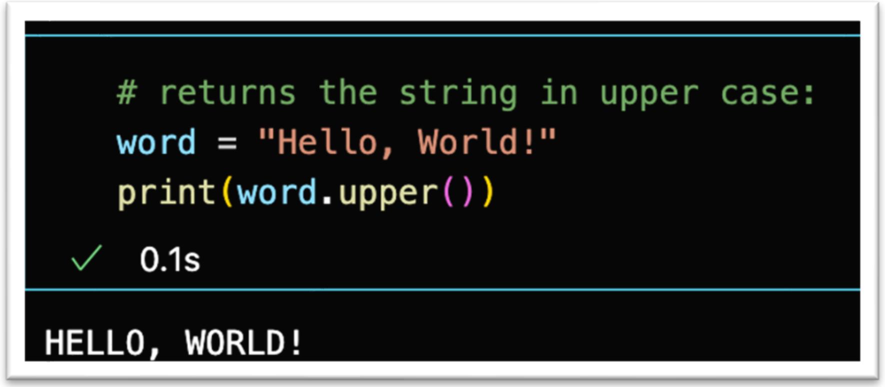


techtalent academy

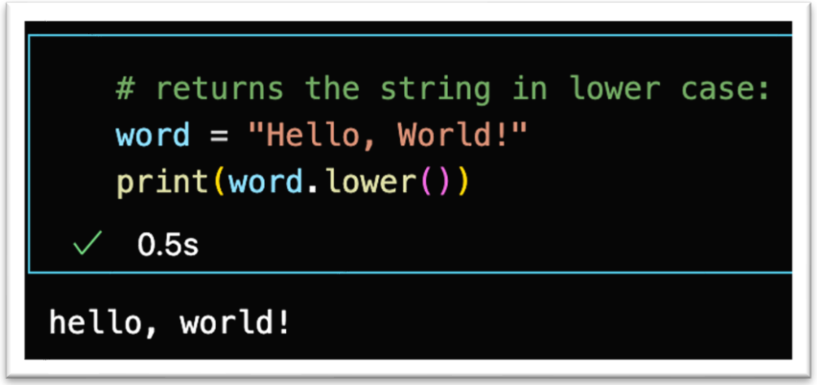
Modify Strings.

Python has a set of built-in methods that you can use to modify strings.

Here are some examples:

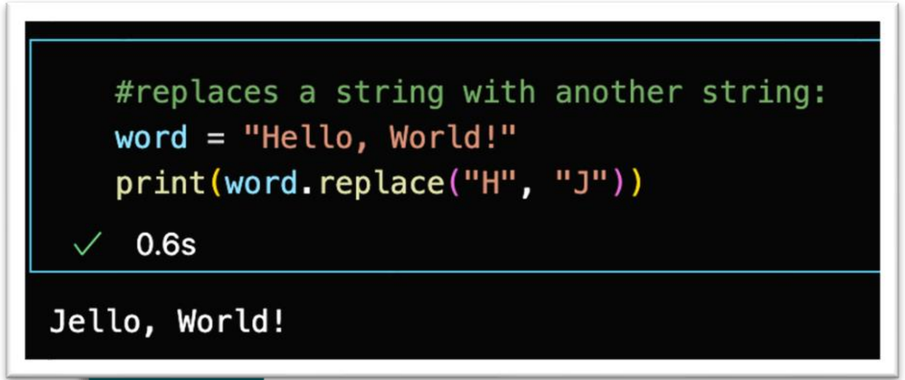


```
# returns the string in upper case:
word = "Hello, World!"
print(word.upper())
✓ 0.1s

HELLO, WORLD!
```

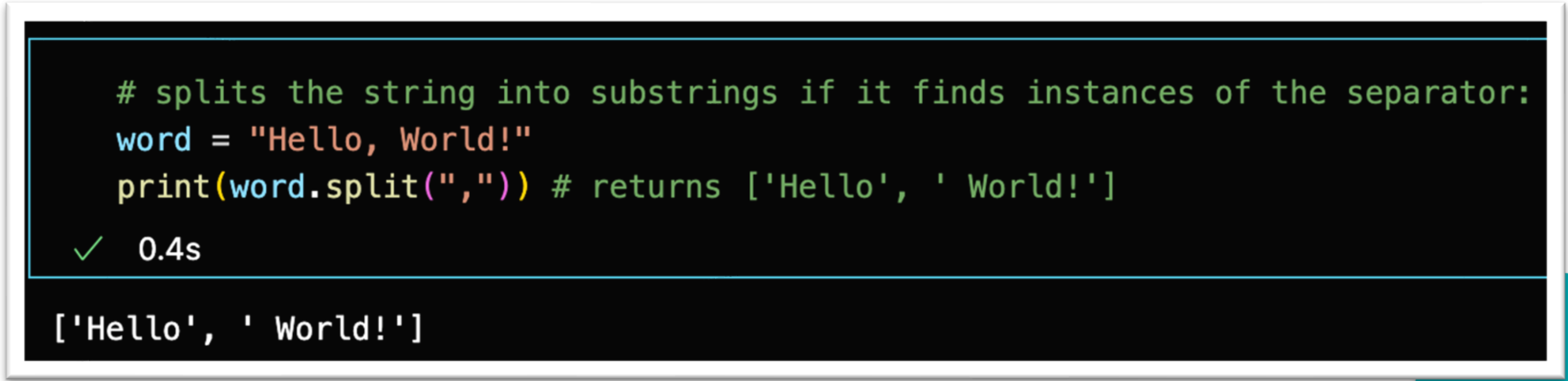


```
# returns the string in lower case:
word = "Hello, World!"
print(word.lower())
✓ 0.5s

hello, world!
```

Methods:
.upper()
.lower()
.replace()
.split()



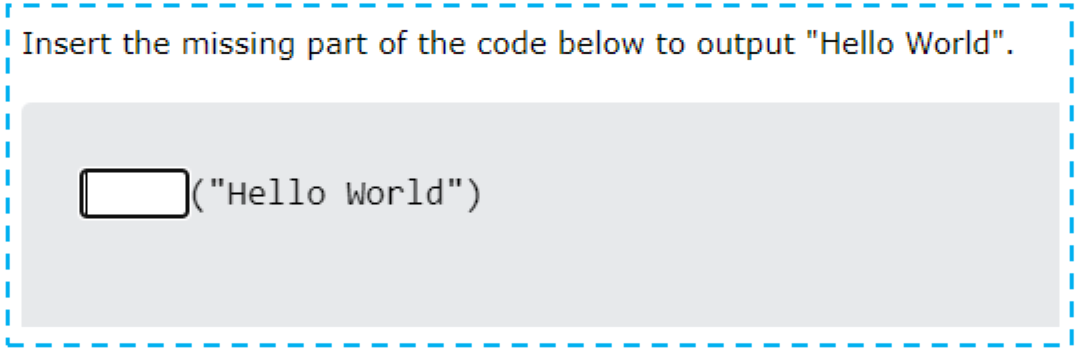
```
#replaces a string with another string:
word = "Hello, World!"
print(word.replace("H", "J"))
✓ 0.6s

Jello, World!
```



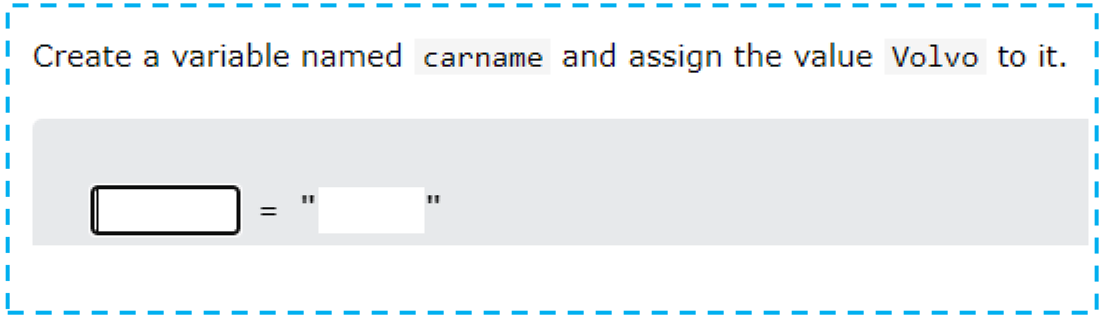
```
# splits the string into substrings if it finds instances of the separator:
word = "Hello, World!"
print(word.split(",")) # returns ['Hello', ' World!']
✓ 0.4s

['Hello', ' World!']
```

techtalent academy

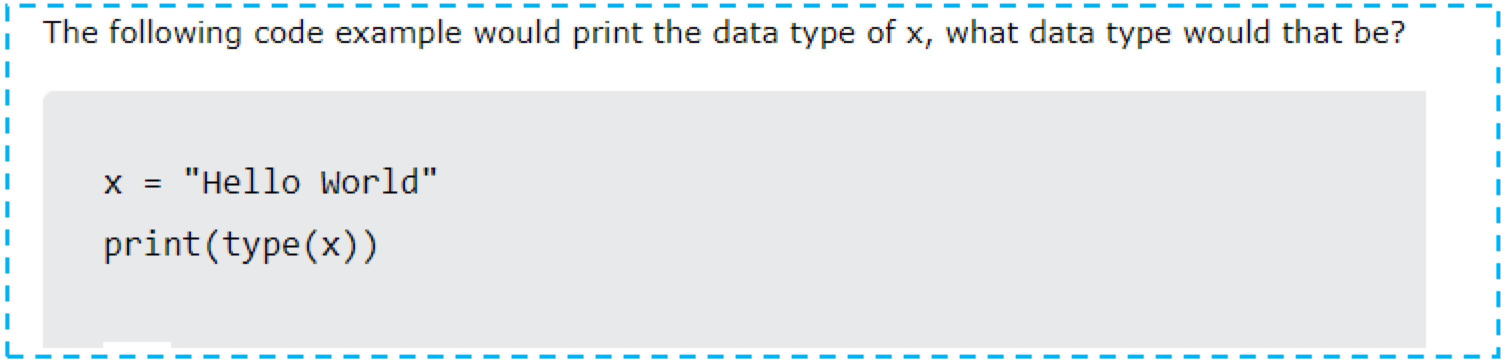
Plenary.


Insert the missing part of the code below to output "Hello World".

```
      ("Hello World")
```


Create a variable named `carname` and assign the value `Volvo` to it.

```
        = "       "
```


The following code example would print the data type of x, what data type would that be?

```
x = "Hello World"
print(type(x))
```