**Bullet Points for Python Basic Section**

**What is Python? Installation and first execution**

* To get started with Python, go to the official website (python.org) and download the version that's right for your system. Then follow the installation instructions.
* You can use Python in two main ways: through the IDLE (Integrated Development and Learning Environment) or the command line. IDLE is a program that comes with Python and provides a user-friendly interface for writing and running code.
* A programming language is a way for humans to communicate with computers and give them instructions. It's like a translator between human language and machine language. Python is a popular programming language that's easy to learn and use.
* When you run Python code, the interpreter reads it and translates it into a series of 0s and 1s that the computer can understand and execute.
* To run a Python command, type it into the interpreter and press enter. For example, you can try basic arithmetic operations like addition and multiplication, or you can print text to the screen using the print() function.

**Variables, creating and running external script, interactive Shell**

* In Python, a variable is a named container that holds a value. When you create a variable, you give it a name and assign a value to it using the equals sign (=). For example, you could create a variable called "message" that contains the text "Hello World" like this: "message = 'Hello World'".
* To create and run an external script in Python, you will need to use a text editor (like Notepad or Sublime Text) to write your code and save it with a .py file extension. Then, you can run the script from the command line by typing "python" followed by the name of the file. For example, if you have a script called "myscript.py", you would type "python myscript.py" to run it.
* The interactive Shell is a command-line interface where you can enter Python commands and see the results immediately. You can use the Shell to test out small pieces of code, try out functions and methods, or just play around with Python. To start the Shell, you can type "python" into the command line and press Enter.
* The Shell is a great place to experiment with Python and learn the language. You can enter any valid Python code into the Shell and see what happens. Just be careful not to accidentally overwrite or delete any important variables or files.

**Comments**

* Comments are notes or explanations that you add to your code to help yourself or other programmers understand what's going on.
* In Python, you create a single-line comment by starting a line with a hash symbol (#). Anything after the hash on that line will be ignored by the interpreter.
* To create a multi-line comment, use a triple quote symbol (single or double) at the beginning and end of the comment. Anything between the quotes will be ignored by the interpreter.
* Be sure to properly end a multi-line comment, or the interpreter may not ignore the rest of the code.
* Comments are a great way to add context and make your code more readable and easier to follow. It's a good habit to add comments to your code, especially in complex or hard-to-understand sections.

**Types of Variables**

* There are several basic types of variables in Python: integers (whole numbers), floats (numbers with decimal points), strings (sequences of characters), and bools (True or False values).
* It's important to choose descriptive and meaningful names for your variables to make your code easy to understand. This will help you and other programmers understand what the variables represent and how they're being used.
* To find out what type of data a variable contains, you can use the type() function. For example, if you have a variable called "age" and you want to know if it's an integer or a string, you can type "type(age)" into the interpreter. This can be helpful if you're not sure what type of data you have, or if you want to make sure you're using the right type of data in your code.
* When you're working with variables in Python, it's important to keep track of their data types and make sure you're using them correctly. Using the wrong data type can cause errors or unexpected results in your code.

**Math Operators**

* In math, there are several common operators that can be used to perform calculations on numbers:
  + The plus sign adds two numbers together
  + The minus sign subtracts one number from another
  + The asterisk sign is used to multiply numbers together
  + The forward slash sign is used to divide one number by another
  + The caret symbol is used to raise one number to the power of another
  + The double forward slash symbol is used to perform "floor division", which rounds down to the nearest whole number
  + The percent sign is used to find the remainder of one number divided by another
* It's important to note that when dividing two integers, the result will be a float number with a decimal point
* Using parentheses in an equation allows you to change the order of operations and specify which calculations should be performed first
* The "modulo" or "remainder" operator takes into account the leftovers after division is performed

**Semicolon and ENTER - assigning multiple values to variables at once**

* To assign multiple values to variables in one line, use a semicolon to separate each value;
* The semicolon acts as an indicator to the interpreter that the instruction has ended;
* You can also use a comma to assign multiple values to variables in one line;
* If you want to assign the same value to multiple variables, you can use the syntax: variable1 = variable2 = variable3 = value;
* This is a faster and easier way to write and read code.

**Assignment operators**

* Assignment operators allow you to increase the value of a variable by a certain amount.
* For example, to increase the value of x by 2, you can write x = x + 2.
* You can also use a shorthand version of this expression, x += 2.
* This operator can be used with any other operator, such as minus, multiply, or divide.
* Try using other shorthand expressions to shorten longer expressions.

**Playing with Strings (Slicing, adding and having fun)**

* Creating a string: assign a variable to a string of characters, e.g. `name = 'Bob'`
* Adding strings: use the `+` operator to add two strings together, e.g. `full\_name = name + ' ' + last\_name`
* Adding spaces: use the `" "` to add a space between two strings, e.g. `full\_name = name + " " + last\_name`
* Creating long strings: use the `\` to indicate that the string
* Slicing strings: use the `[start:end]` syntax to access a range of characters in a string, e.g. `name[0]` to access the first letter of the `name` string
* Counting from the back: use negative numbers to count from the back of a string, e.g. `name[-1]` to access the last letter of the `name` string