**Statement**

*One line of code*

**Expression**

*An equation that the must be solved by the interpreter before continuing*

**Substitution**

*When the interpreter replaces a variable, expression, or function with its value*

**Variable**

1. *Has an identifier (name)*
2. *Contains a value*
3. *Assignment Operators, like ‘ = ‘ are used to assign a value to an identifier*
   1. *variableName = value*
4. *When the Python interpreter reads the identifier, it substitutes it with its associated value*

**Data Types:**

1. *Integer – A whole number*
2. *Float – A number with a decimal*
3. *Boolean – ‘True’ or ‘False’*
4. *String – A string of characters (letters inside of “quotes”)*

**Conditional** **Statement**

1. *Statements that run code blocks when their condition is Not False*
2. *A code block comes right after a conditional statement*
   1. *It is indicated by every statement in the code block being one*
3. *Always have a:*
   1. *Keyword (****if****,* ***elif****,* ***while****, etc.)*
   2. *Condition*
   3. *Colon* ***:***

**Condition**

1. *An equation or value that can be ‘True’ or ‘False’*
2. *Examples: “5 > 2”, “x == False”, “len(team) >=6”*
3. *Uses comparators: >, <, >=, <=, ==*
4. *Has extra keywords that modify the condition(s): ‘not’, ‘and’, ‘or’*

**Function**:

1. *A special command that always ends with parenthesis* ***()***
2. *You can put extra data in the parentheses* ***()***
   1. *print( “this string will be printed to the console” )*

**Assignment Operators**

* *Assigns the value of the right side of the operator to the identifier on the left side*
  + *x = 5 # ‘x’ is now holding the value ‘5’*
* *Always has an equals symbol =*
* *Putting an Arithmetic Operator in front of the equals* ***=*** *performs that operation on the value already assigned to that identifier*
* **= += -= /= \*=**

**Arithmetic Operators**

* Tells the interpreter to perform a math equation
  + *x = 6 + 5 # Adds 6 and 5 together, then stores that value into ‘x’*
* Performs the selected mathematical operator on the two values to the left and right of it
* **+ - / \* ^**

**Comparison Operators**

* Compares each value to the left and right of it, then substitutes the expression with ‘True’ or ‘False’
  + x = 6 > 5 # 6 is greater than 5, so the value of ‘x’ will be ‘True’
* **== > < >= <=**

**Logical Operators**

* In Python, these operators are words that can change how expressions are solved
* ‘not’ inverts a value. If the value is ‘True’ it becomes ‘False’ and vice versa
  + x = True  
    y = not X # The value is y is ‘False’, because the value of y is ‘True’
* ‘and’ chains two comparisons together, and substitutes itself with ‘True’ if both comparisons are also ‘True’
  + x = 6 > 5 and 3 < 5 # x is ‘True’ because 6 is greater than 5 AND 3 is less than 5
* ‘or’ chains two comparisons together, and substitutes itself with ‘True’ if *either* comparisons are also ‘True’
  + x = 6 > 5 or 3 > 5 # x is ‘True’ because 6 is greater than 5, even though 3 is not greater than 5
* **not and or**