Python Study Sheet — Russell Overton

**Basic Coding (typically roughly the same across languages, minus stuff like dynamic typing, print statements, and types of errors, like how lower level languages have stuff like memory errors)**

* **print**("") prints strings, where the text is inside the parentheses and the quotation marks A black background with green and purple letters

  AI-generated content may be incorrect.
* There are a few different types of errors. The two most common and likely to pop up in python are **syntax** and **logical** errors. Syntax errors are essentially just typos. They often occur due to missing parentheses or periods or semicolons. Logic errors are when the program runs but it produces an unintended result. This is because of programmer error.
* The data types are numerous, but the most common in our use case are **int**, **boolean**, **double**, **float**, **string**, and **list**. **Int**, **double**, and **float** are all types of numerical variables, booleans are binary, strings are just anything in quotation marks, and lists are lists.
* You declare a variable by doing **[variable\_name] = [data]** **A black rectangular object with white text

  AI-generated content may be incorrect.**
* python is **dynamically typed**, so you don't have to declare what type of variable a variable will be and it can change
* one equal sign is to set something to a value, two equal signs is to check if equal A black background with white and green text

  AI-generated content may be incorrect.

for example, these are different statements, where the second one evaluates to true.

* you can do **str()** or **int()** etc to convert a data type to another
* use **#** to comment your code. make sure to comment it enough to make it clear what it does to a reader A black background with white letters

  AI-generated content may be incorrect.

**Packages**

* **import [blank]** brings in **packages**, which allow you to use other peoples' functions and apis etc (simplified)
* If we import turtle, for example, the way to call stuff in turtle is turtle.[function]
* You can specify what you'd like to import. For example, **from** turtle **import** right just imports the one command from turtle, seen here: 

**Functions**

* define **functions** by writing "**def**" and then whatever the function name is. After that, you define the **parameters** that need to be inputted to the function, then begin the function with **:**

For example: A black background with colorful text

AI-generated content may be incorrect. is the skeleton of a function, where def is the beginning of the function statement, which says that we’re going define a function. Function is just the name of the function. Parameter is where you put parameters, and the return statement is where the function returns something.

* There are a few types of loops in python, but the two that i think are most common are "**for**" and "**while**" loops. For loops are best used when you **know how many iterations** you want to do something over, while "while" loops are best suited for when you **aren't sure of how many loops something needs to do**. They both first check to see if the condition is true and then run the body of the loop if it still is.
* The way to call functions that you've defined in python is with **function(parameters)**
* **if** statements A black background with white text

  AI-generated content may be incorrect. check if a thing is true and then run the code inside the statement if it is, where in the image, condition is the thing we check and do\_thing is the code that would then run