Lesson 1: Introduction to Python and Programming (Refresh) *Programming Basics I*

- Python is a programming language that is executed or performed line-by-line.
- Python is a widely used language and can be used for many different purposes. It is simple and clear, without being overly verbose.
- On PyKidz, you will be learning more about the general concept of the different aspects of Python and general programming concepts.
- Computers do *exactly* what they're told to do, and only that. In order to effectively write computer programs, we must think like a computer, which is called writing or developing an algorithm. "Programming" is actually translating this algorithm into a language the computer understands, which is python in our case.
- Some important qualities you will develop as a programmer include: creativity, asking questions to others, persistence, and a new way of thinking!
- To practice this, take a couple minutes to split an everyday task into individual parts that an alien could understand. For instance, write directions of how to make a peanut butter and jelly sandwich in a perfectly clear way so that no computer or alien gets confused. *timer starts*

In any python file (one that ends with ".py") you can just type this into the file, save, then run it!
print("hello world"**)**
maybe this is like lesson 0, then 1 starts after??

Lesson 2: Basic Data Types *Data Types/Structures I*
(introducing data types after the general programming thinking structure?)

- BASIC DATA TYPES:

  First, we will be teaching you some of the basic data types in python. What is a Data Type? Well,it's a special category that tells the computer how to treat a piece of information. Think of it as having different kinds of toys in your toy box, some are dolls, some are cars, and others are legos. Just like how you would organize your toys, computers organize information into different types. First we will talk about the most basic ones.

  - Integer('int') = represents whole numbers without any decimal points. When thinking an integer think about counting marbles or apples in math, you can only have a whole number.

  - Float ('float') = represents floating -point numbers, which include decimal points. Think about it like this, it's like counting slices of pizza. Each half of a pizza is ½ which would be represented as 0.5.

  - String ('str') = Represents sequences of characters, enclosed in single or double quotes. Strings are basically any word! Can be your name, a name of a friend or pet, or literally anything! Computers call these words, "strings".

  - Boolean ('bool') = represents true or false. Think of this like a light switch, it's either on or off and helps determine whether a light is turned on or off, where ‘true’ would equate on and ‘false’ would equate off. This helps the computer make decisions. They must be capitalized to hold any significance in python.

- POTENTIAL QUIZ/REINFORCMENT QUESTIONS:
    - WHich of the following is an example of an ‘int’ :
        - x=5 <-
        - x=3.14
        - y = True
        - y= ‘five’
    - Which of the following is NOT a data type in Python
        - ‘Int’
        - ‘Float’
        - ‘Char’ <-
        - ‘Str’

Lesson 3: Syntax & Math (applying the last lesson??) *Syntax I*

- Circling back to the “hello world” statement,
- It’s important that the names of these variables (identifiers) always clearly correspond with what they represent. These should be short and clear, consisting of letters, numbers, and/or underscores. A good name for a variable representing how many times it takes Nick to score a point while playing basketball is numShots, count, nickBaskets. Keeping the names short, rather than numberOfTimesNickShootsBeforeMakingABasket keeps your code easy to read.
    - identifying which variables are named well
- Math! Different symbols carry different significance in programming languages:
    - %
    - /
    - ^
    - *

Lesson 4: Advances Data Types *Data Types II*

- After lesson 1 I hope you now know what a Data Type is and remember some of the ones discussed. If not, don’t worry! You can feel free to go back and review some more!
- For lesson two, we are going to talk about some more, slightly more complex data types, but don;t worry! We will guide you through these and explain this in a way we hope you understand!

- List (‘list;): Represents an ordered collection of items, which can be different data types. Think a list as a toy box, and the toys that goo in the toy box would be other data types. The toy box can hold all the toys together in a specific order, for example you can order it for example in alphabetical or numerical order!
    - Ex: ‘my_list’ = [1,3.14, “three”, True]’
    -

- Tuple (‘tuple’): A tuple is very similar to a list, the only different is that you can’t add or get rid of anything in it. Think of it as a bag of toys you want to take on a roadtrip, and you cannot change once you have packed it. Your mom says you can’t take anything out because you might lose one of your toys. So you are not allowed to take anything out of this back and you also can’t add anything once packed.
    - Ex. ‘fav_toys’ (“dinorsaur”,”doll”, “race car”)

- Dictionary (‘dict’): Represents a collection of key-value pairs. Think of a dictionary as a matching game where each one of your toys has a special friend that goes along with it. So if you have a toy, let’s say, Barbie, then Barbie’s friend is Ken, or if you have a toy Lightning McQueen, then Lightning McQueen’s friend is Mator.
    - Toy_friends = { “Barbie” : “Ken”, “Lightning_Mcqueen”: “Mator”}
    -
- Set (‘set’): A set is an unordered collection of unique elements.
    -

Lesson 5: Object Oriented Programming, Polymorphism: *Programming Basics II*
Polymorphism:

- One benefit of polymorphism is that code can be reused by different methods!
- A method is a function that only runs when it is called. These consist of a definition, written “def” in python, a name, and a set of parentheses before a colon. Methods are *called*
    - Methods can be called from other methods, or from themselves. Generally the first method that automatically runs upon the program beginning is called “main”.
- Parents: a
- The pb&j sandwiches are an example of procedural decomposition! By splitting an activity into its finer parts, we can recognize spots where the program is redundant, encouraging us to determine how we can manipulate it for reusability
- Examples & Identification given certain relationships between objects:
    - A dog is an instance of a pet and a pet is an instance of an animal and each level in these has certain attributes, like
    - Animal: Number of legs, color, age
    - Pet: Name, Adoption date
    - Dog: Bark sound, Leash color, Favorite toy
- These are just to get beginners thinking about the relationship between objects in their lives through a programming lense

Further classes:
--data structures
--binary numbers (fun tricks?)
--abstraction
--modular math, explaining remainders and the different symbols, syntax
--algorithms
--extracting data and splitting it in a way a computer can use (mc, no care about syntax)

--casting
--pseudocode