Vocabulary By Mission / Assignment

Mission 1 and Mission 2 – Welcome & Introducing CodeX	
bug	When your program doesn’t do what you intended it to do
debugging	the process of understanding what the computer is actually doing and then changing the code to do what you want it to do
CPU	Central Processing Unit or the brain of the computer
peripheral	A device that interacts with the CPU (common peripherals are LED lights, display screen, buttons, mouse, keyboard, and printer)
Mission 3 – Light Show	
RGB	Red, Green, Blue; the colors that make up a single pixel on the screen
sequential	Executing code line by line, one after another, in order
literal	a specific value, like 1 or “hello”
variable	a name you assign to some data that you use in code instead of the literal, or actual values
assign	Bind a name to a value; give a variable a value
Mission 4 – Display Games	
argument	Passing data to functions (information a function uses to complete its task)
integer	A whole number that can be positive, negative or zero
string	A sequence of characters, like words or sentences
Conversion function	a built-in function that converts a value to a different (and specific) data type
branching	Decision points in code; a condition
selection	Decision points in code; a condition – this isn’t in the documentation but is used in AP CSP
boolean	True or False data type (values that can be True or False)
indentation	Structuring blocks of code in Python; statements ending with a colon (:) execute the block of code indented four spaces beneath it
Mission 5 – Micro Musician	
readability	Making code easy to understand for humans.
comments	Notes in code that are ignored by the computer but can explain what the code does These vocab words are not specifically in the Mission instructions, but are included in the warm-up and can be added either in warm-up or wrap-up. Should be covered for the AP exam
analog	Smooth and continuous signals that represent a quantity, like sound waves
digital	A numerical representation of an analog signal, represented in increments

Design Process and Flowcharts	
Design process	a tool that helps you break down large projects into smaller, easier-to handle stages
algorithm	a sequence of steps for completing a task (step by step process)
flowchart	a diagram that uses shapes, lines, and arrows to sequence steps; a visual representation of the input, output, decisions, and actions that take place within a program
Mission 6 - Heartbeat	
loop	Repeats a block of code, subject to a given condition.
While loop	Repeats a block of indented code as long as the condition is true.
Infinite loop	A loop that never ends because the loop is always true.
iteration	The repeating portion of an algorithm; code that repeats until a given condition is met or a specified number of times.
Increment (a counter)	Increasing a variable by a specific amount. Often counters are incremented by one: **count = count + 1** , like a counter, but the value can be any literal number (or constant).
Decrement (a counter)	Decreasing a variable by a specific amount. Often counters are decremented by one, like a countdown: **count = count - 1** , but it can be any literal number or constant.
Define and Call Functions Lesson	
Abstraction	the process of taking away or removing characteristics from something in order to reduce it to a set of essential characteristics
Function	a named set of instructions that accomplishes a task
Mission 7 – Personal Billboard	
Comparison operator	Operators that let you compare two values; the result is True or False. Comparison operators include: ==, <, >, <=, >=, !=
Index	A number that keeps track of what choice should be displayed.
Nested Condition	Another if statement that is part of (embedded in) the block of code in an if statement (an if statement within an if statement).
List	A sequence of items you can access with an index.
Lists Practice #1	
List	an ordered collection of elements
Index	a common method for referencing the elements in a list or string using numbers
Element	an individual value in a list that is assigned a unique index
List Length	how many elements it contains. Lists can grow or shrink as elements are added or removed. You can calculate the current length by using the function: ***len(list_name)***
Mission 8 – Answer Bot	

Range	A sequence of numbers you can iterate over. You must provide at least the stop (or last) number in the sequence. Optional: you can provide the start (or first) number in the sequence and also the step, if other than increasing by 1.
Constant	A named value that doesn’t change during the run of the code. By convention, constants are represented with ALL CAPS
Types of Division	
Decimal (or real number) division	A regular mathematical division problem, where the answer is always a decimal (or real number) even when the divisor goes into the dividend evenly
Integer division	The whole number from a long division problem – the number of times a divisor goes evenly into a dividend
Modulo (or modulus) division	The remainder of a long division problem – the amount leftover from a divisor and a dividend
Mission 9 – Game Spinner	
Logical Operator	Operators that handle combinations of Boolean results; not, and, or
Function	A named chunk of code you can run anytime just by calling its name; also called a procedure
Simulation	Code that builds a *model* of something, and lets you play with that model. Simulations let you explore "virtual" situations, both realistic and imaginary, that might be difficult or impossible to do in the real world.
Parameter	A local variable in a function that receives a value passed into the function when it is called; information the function needs to complete its task
Argument	The value passed into a function – information the function needs to complete its task. An argument can be a literal value, a variable, or an expression.
Local variables	Variables defined inside a function, and can only be used within that function.
Mission 10 – Reaction Time	
Argument	Review from previous mission
Computer clock	Electronic clock circuits; the heartbeat of the computer. The tick of the clock moves through the code one line at a time. It is also used in the sleep function, scheduled activities within the CPU, and everything timing related on the computer.
Monotonic	Always increasing (a computer’s electronic clock is not monotonic; like an odometer it will wrap-around once the highest value is reached)
Mission 11 – Spirit Level	
Accelerometer	A device that measures proper acceleration; a sensor chip that detects motion, impacts, and orientation
Tuple	An *immutable* sequence of items that you can access with an *index*, or a list with values that don’t change. A read-only version of a list.
Mission 12 - Night Light (review from Mission 5)	

Analog	Infinite variation in something, like hot to cold or light to dark; smooth and continuous signals that represent a quantity, like sound waves
Digital	A numerical representation of an analog signal, represented in increments
ADC	analog to digital conversion
Traversing a list	
Traversing	traveling or traversing through a list one element at a time, in order, starting with index 0 (first element) and going through to the last element (index len-1)
Sequential (M3)	Executing code line by line, one after another, in order
Selection (M4)	Decision points in code; a condition – this isn't in the documentation but is used in AP CSP
Iteration (M6)	The repeating portion of an algorithm; code that repeats until a given condition is met or a specified number of times.
Traversing list program – review from previous lessons	
Traverse	traveling or traversing through a list one element at a time, in order, starting with index 0 (first element) and going through to the last element (index len-1)
	Review **sequential, selection,** and **iteration** from previous lessons (Mission 3, 4, and 6)
Functions, parameters and local variables – review from Mission 9	
Function	a named set of instructions that accomplishes a task (A named chunk of code you can run anytime just by calling its name; also called a procedure)
Parameter	A local variable in a function that receives a value passed into the function when it is called; information the function needs to complete its task
Argument	The value passed into a function – information the function needs to complete its task. An argument can be a literal value, a variable, or an expression.
Local variable	Variables defined inside a function, and can only be used within that function.