# CodeBot Vocabulary By Mission

| Mission 1 -- Welcome | |
|---|---|
| Browser | Software that displays web pages |
| Cloud | A place to save files and data through the Internet |
| Objective | The steps in the mission; has a goal to accomplish |
| Text editor | Where you type the code |
| Code | Instructions to the computer |
| Toolbox | A place in CodeSpace to keep information you learn about programming concepts so you can use it later when you need the information |
| Simulation | A 3D environment that lets you see the robot move and interact in a virtual world |

| Mission 2 – Introducing CodeBot | |
|---|---|
| CodeBot | A computer on wheels with lots of sensors and controls built-in |
| Peripherals | Devices that give input or output to CodeBot (some CodeBot peripherals are LED lights, speaker, motors, line sensors, proximity sensors, an accelerometer, and pushbuttons) |
| Motors | Programmable electric engines; powers the wheels |
| LEDs | Light emitting diodes; tiny and efficient electronic components that produce light |
| Wheel encoders | Discs that rotate, counting the invisible IR light beam pulses through its slots |
| Static electricity | A charge that can build up and causes a jolt and spark when grounded |
| Comment | Code that doesn't get run (more information in Mission 3) |
| Import | Provides access to a module (or library) of built-in Python functions to use in your code |

| Mission 3 – Time and Motion (Objectives 1-6) | |
|---|---|
| Physical computing | Writing code (instructions) for a physical device, like CodeBot or cars |
| Editor shortcuts | Keyboard hotkeys to write code faster; combinations of keys which complete a task |
| CPU | The "brain" of the computer that executes your code; the Central Processing Unit |
| Debugging | The process of understanding what the computer is actually doing and then changing the code to do what you want it to do |
| Delay | Functions that slow things down, like sleep(); the module must be imported first |
| Blocking functions | Functions that pause program execution; no other code will run during the pause |
| Literal | An actual value, like 1 or "hello" or True |
| Variable | A name to which you assign some data, any type of information your program uses; must be defined before it is used |

| Boolean | A value that is True or False |
| --- | --- |
| Argument | Passing data to a function, determined by the position in the list when the function is called; arguments can be literal values, like True, or variables, like delay |
| Binary | How a computer deals with digits; electrical connections, like switches, that are either on or off (2 states) |
| Byte | 8-bits of binary data |
| **Mission 3 – Time and Motion (Objectives 7-9)** | |
| Comments | Notes in the code about what you are doing; increases the readability of code and is meant for humans, not the computer (they are not instructions to the computer and are not executed) |
| Whitespace | Adding blank lines and space around symbols to make the code more readable (ignored by Python, non-executable) |
| Algorithm | A precise sequence of instructions that the computer can follow exactly, one step at a time, to complete a task or solve a problem |
| **Mission 3 – Time and Motion (Objectives 10-11)** | |
| Control flow Branching | Decision points in code; code will take a different branch or path depending on a condition |
| Condition | A Boolean value (True or False), often the result of a comparison operator like <, > or = <br> Use an if statement, optionally followed by an elif or else, for branching |
| Indenting | A way to structure blocks of code by offsetting a block of code four spaces; blocks of code are indented following a statement with a colon (:) |
| **Mission 4 – Animatronics (Objectives 1-5)** | |
| Loop | Changing the flow of the code by repeating a block of code, subject to a condition |
| While condition | A statement that tells Python to repeat the block of code as long as the given condition is true |
| Infinite loop | A loop that never ends because the condition is always true |
| Updating a variable | Assign a new value, based on the old value of the variable. |
| Increment | Update a variable by adding one (or any specific number) to the old value |
| Single equal (=) | Assignment – used to assign a value to a variable |
| Double equal (==) | Comparison operator to determine if two objects are the same |
| Break | Exit the nearest enclosing loop |
| **Mission 4 – Animatronics (Objectives 6-12)** | |
| Debounce | Reset the internal status of a button so the press isn't counted twice |
| While loop | A loop that iterates, or repeats, while a condition is true. `count = 0` `while count < 10:` In this |

| | example, the loop will iterate 10 times, with count having the values 0 through 9. When it becomes 10, the condition is no longer true and the loop ends. |
|---|---|
| Parameter | A list of names declared in a function definition that receive values when the function is called and act like local variables in the function |
| Random number | When using randrange, a range of numbers is given (start, stop). The random number will include start but will be less than stop. |
| Function | A named chunk of code you can run anytime just by calling its name; reuse code without retyping it |
| **Mission 5 - Fence Patrol** | |
| Line sensors | Photo reflective sensors that detect lines and boundaries beneath your 'bot |
| API | Application Programming Interface – the details of how your program interacts with different services it needs |
| Analog | Infinite variation, like from dark to light or cold to hot |
| ADC | Analog to digital converter |
| REPL | Read Evaluate Print Loop – the command line that lets you type Python statements directly and observe what happens |
| DRY | Don't Repeat Yourself – never write the same code twice |
| Return statement | Exits the function and sends a value back to the code where the function was called |
| **Mission 6 – Line Follower** | |
| list | A sequence of items you can access with an index |
| tuple | Read-only form of list |
| Or (Logical operator) | Multiple conditions to compare, testing if either one or the other or both is true |
| Hard coded values | Specific numbers used in code that can be replaced with a variable or constant |
| globals | Variables defined outside of a function; they are available during the entire program and can be accessed throughout the entire program |
| locals | Variables defined inside a function; they only exist while the function is running and can only be accessed in the function |
| Int (integer) | A value that is an integer; designated by int in Python; can be positive or negative |
| Float (decimal) | A value that is a decimal, also known as a floating point; can be positive or negative |
| auto-calibrate | Use CodeBot sensors to automatically adapt to its environment by detecting lines and objects and setting parameters like is_reflective and thresh. |
| **Mission 7 – Hot Pursuit** | |
| Proximity sensors | Infrared (IR) sensors that can detect nearby objects based on the reflected IR light |
| Detection sensitivity | How much light is needed for the proximity sensor to detect an object (from 1 to 100) |

| Emitter power level | The brightness of CodeBot's IR flashlight, with settings from 1 to 8 (high power) |
|---|---|
| Not (logical operator) | A special kind of logical operator that needs only one Boolean operand, and inverts it; it can be used to toggle a Boolean variable |
| **Mission 8 – Navigation** | |
| Wheel encoders | A disc with slots that rotates with a wheel so that an IR light beam can pass through its slots. The pulses of light can be counted to see how much the wheel has rotated. |
| state | Property of an object; for example True or False. The state can be stored in a variable, so the current state can be compared to the previous state. |
| speed | Distance / Time (can be any measurements; we will use cm/s - centimeters per second) |
| Iterative process | Repeatedly taking small steps to build a whole solution |
| Closed loop control | Automates control of a system by sensing the output state and comparing it to the desired state (input). |
| Feedback loop | Continuously adjusts the system to keep the error, or difference between input and output, close to zero. In our mission, the feedback comes from the encoders, the input is the desired speed and the output is the actual speed. Disturbance can be friction, surface, etc. |
| Breakpoint | A marker you can place on any executable line of code that will cause the debugger to stop. |
| **Mission 9 - All Systems Go!** | |
| Under load | Batteries are being used to power a peripheral, like turning on LEDs or running motors |
| User interface | The UI, the part of the computer that humans interact with directly. On the 'bot it can be buttons and LEDs. |
| Ambient | Surroundings |
| Baseline data | Starting point used for comparison; original data |
| Deadband | In a control system, the range or band of input values where the output doesn't change; like a threshold. |
| Accelerometer | A tiny chip that measures the force of acceleration in 3 directions: x, y and z |
| MEMS | Micro-Electro-Mechanical System; a chip with tiny silicon structures inside that really move, with electronic components to sense them. |