# Lesson 3.01: Built-In Functions ## Learning Objectives Students will be able to... * Define and identify: **function, arguments, calling, importing, returning** * Call the built-in `randint` function, using arguments * Utilize code other people have written in the Python documentation * Understand the difference between printing and returning ## Materials/Preparation * [Do Now] * [Lab - Magic 8-Ball] ([printable lab document]) ([editable lab document]) * Associated Reading - section 3.1 of Book * Read through the do now, lesson, and lab so that you are familiar with the requirements and can assist students. * **Microsoft Learn - Parameterized Functions Video** [![Functions Video](https://img.youtube.com/vi/sKW-zdYZNX4/0.jpg)](https://www.youtube.com/watch?v=sKW-zdYZNX4) ## Pacing Guide | **Duration** | **Description** | | ---------- | ----------- | | 5 Minutes | Do Now | | 10 Minutes | Lesson | | 35 Minutes | Lab | | 5 Minutes | Debrief | ## Instructor's Notes ### 1. Do Now * Give students 3-4 minutes to follow the instructions on the Do Now page. * Debrief the answers to the questions on the Do Now by calling on students to respond. ### 2. Lesson #### Build Your Own Blocks (Snap) vs Functions (Python) * Ask students to recall how they built custom blocks in Snap! ![Snap Custom Block](https://i.ytimg.com/vi/Bbl2fh3igQ4/maxresdefault.jpg) * **Function**: a named sequence of statements. You can use functions to perform complex calculations, graphical operations, and various other purposes. When you define a function, you specify the name and the sequence of statements. Later, you can “**call**” the function by name. * In SNAP! functions are blocks #### Function Contract A function definition is like a contract: you tell the programmer what elements the function expects (name and type of arguments) and the function will perform its purpose. It is good practice to use a comment to specify the purpose and contract of a function, including the type of value it returns, if it returns a value. ```python * Name: * Purpose: * Arguments: * Returns: ``` * Explain that we have already gotten used to **calling** functions like `type()` and `print()`. * Ask students how they would create a random number generator. * Sounds hard! Luckily someone has already done that: the random library (essentially a bunch of code written by someone else) which has many associated functions. #### Random in Snap ![Random in Snap](http://bjc.berkeley.edu/bjc-r/img/lab-10/random-tree-buggy-code-snap.png) #### Back to the Do Now * Remind students what they saw in the Do Now - how to get a random integer: randint(0, 10). * Identify the 0 and 10 in this example as **arguments**, or values passed into the function. * Ask students what the argument is when we use `print` or `type` * `randint` gives back a value that you might want to store - this is called **returning**. If nothing is given back, the return value is `None`. #### More on Function Contracts * Functions have a contract: you write down the name, purpose, arguments with their type, and the return type expected. * Ask students what the contract of `randint` is. ```python * Name: `randint` * Purpose: generate a pseudo-random integer N such that a <= N <= b * Arguments: 2 values of type integer: a and b * Returns: integer ``` * Since `randint` is written by someone else there is a place where that contract is written out - **Documentation**. Have students begin the lab, which will instruct them to find the Python documentation for the random library. ### 3. Lab * Students look through `random` library documentation, practice importing different random functions and using them. * Create a Magic 8-ball program using a list and `randint`. **Video Explanation of the Magic 8 Ball** [![Magic 8 Ball](https://img.youtube.com/vi/gMSPH1Cnwwo/0.jpg)](https://www.youtube.com/watch?v=gMSPH1Cnwwo) ### 4. Debrief/Exit Ticket * In their notebooks, have students right down 2 things they learned today to reinforce learning. ## Accommodation/Differentiation If students are moving quickly, find another library to import from (see **bonus** in the lab) OR allow students to move on to creating their own functions. ## Forum discussion [Lesson 3.01: Built-In Functions (TEALS Discourse Account Required)](https://forums.tealsk12.org/c/2nd-semester-unit-3-functions/lesson-3-01-built-in-functions) [Do Now]:do_now.md.html [Lab - Magic 8-Ball]:lab.md.html [printable lab document]: https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/3_unit/01_lesson/lab.pdf [editable lab document]: https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/3_unit/01_lesson/lab.docx