LESSON _1: VARIABLES IN PYTHON

OBJECTIVES

Students will be able to:

1. Identify a variable as a way to label and reference a value in a program
2. Use variables in a program to store a piece of information that is used multiple times
3. Reassign variable values

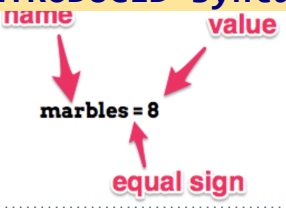
STANDARDS

Computational Thinking: 7-8.CT.1, 7-8.CT.

VOCABULARY:

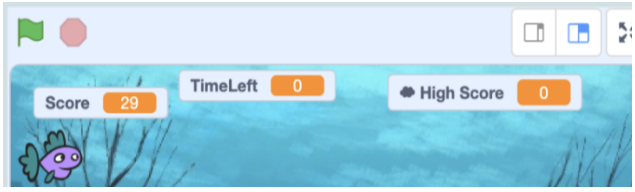
Variables, initialize, assignment statement, reassign.

INTRODUCED syntax:



WARM UP:

Activate students' prior knowledge of block code



Display the above image from the fish chomp game we created last year and ask students:

- How many variables did we create?
- What is the name of each variable? What is the value that is tied to each variable?

LESSON DEVELOPMENT (WHOLE GROUP)

As you recall from Scratch/Game lab, the variable name is used to reference that stored value within a computer program. The same concept applies in Python, you can think of a variable as a label that has a name on it, which you tie onto a value:

The label has the variable name my_int written on it, and is tied to the integer value 103204934813.

The phrase **my_int = 103204934813** is an **assignment statement**, which consists of a few parts:

- the variable name (my_int)
- the assignment operator, also known as the equal sign (=)
- the value that is being tied to the variable name (10320493481
- As soon as we set a variable equal to a value, we **initialize** or create that variable. Once we have done that, we are set to use the variable instead of the value. **For example, print(my_int) ---> 103204934813**

  Why Use Variables? If we need to use this number multiple times 103204934813, we store it in a variable rather than continuously retype the long number over and over again. Instead, we use something that's easy to remember like the variable my_int

➔ How do we name variables? Naming Conventions in Python

Variable names should indicate their purpose

- Can contain **letters, numbers, and underscores**
  - Use underscores to separate words
- Python is **CASE SENSITIVE** (e.g. Num_kids vs. num_kids)
- **No spaces, symbols, or keywords**:
  - Keywords: **print, False, True, else, for, while**
- **Can't** start with a number **(e.g. 1Game vs. Game1)**

✔ CHECK FOR UNDERSTANDING

Which variable names are valid? Explain

1. hello-world
2. print
3. number_1_fan
4. 1_number
5. a7a777dddd7d7d7d7
6. #mfaProud
7. this_name
8. while
9. I love coding
10. _CoolGuy_

UNPLUG ACTIVITY (WORK IN PAIRS)

Exploration of Reassigning Variable Values (Code Prediction)

Input	What is the output?	Justify
x = 8 x = 10 print(x)		
x = 20 x = x + 1 x = x + 3 print (x)		
x = "Sammy" print(x)		
x = 76 x = "Sammy" print(x)		
Challenge x = 5 x = "hello" print("Oh, "+ x +"!")		
x = 5 y = 10 print(x + y)	Segway to our next lesson: data types	
x = “hello “ Y = “world” print(x + y)		
x = “5” y = “10” Print (x + y)		

SHARE OUT (WHOLE GROUP)

- Have students share out their predictions
- Confirm students' predictions by running the code on the smart board using https://replit.com/~

✔ HOMEWORK _ASSESSING QUESTIONS

1. Can we reassign a different value to the same variable?
2. What would be the output?

   Challenge
3. What is the difference between 5 and "5"? **Segway to our next lesson: data types**

SUMMARY

BIG IDEA

- Variables are used to store a piece of information that is used multiple times. We can reassign variable values –>the output will be the value of the most recent assignment.

RESOURCES

Code.org/CSD