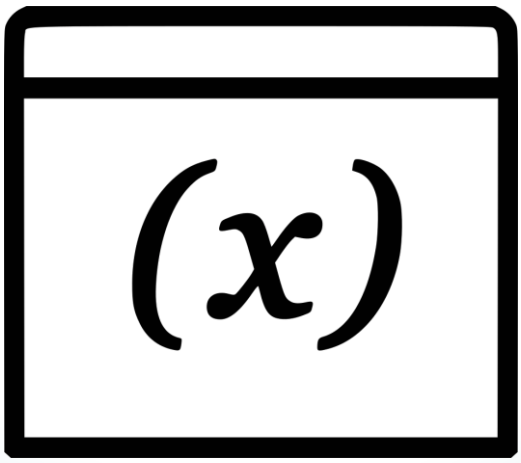
Computer Science
Programming Fundamentals

Variables



💡 Variables Intro

```
print("Joe")
```

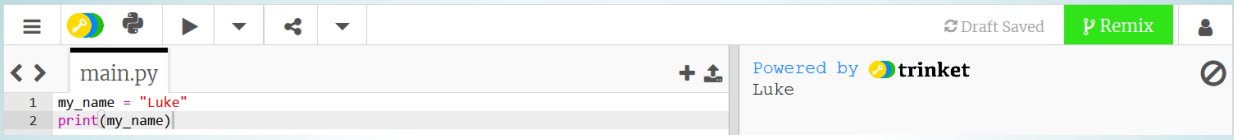
When we run the code above, Python will output “Joe” and then forget it. However, we often need Python to remember what we tell it. **Variables** tell Python to remember something so we can use it later.

When we use variables, our programs become flexible, and we don’t need to write as much code.

Variables are called variables because they can vary, or change. You may have heard about **constants**. Constants do not change. See the glossary for more information.

Variables pt 1

- Run the code in the example.



Q: What is = doing in this code?

A: It is telling Python to remember that `my_name` is `"Luke"`. The **value** of `my_name` is `"Luke"`.

We don't need quotes around `my_name` because it is a variable.

- Run:

```
print(my_name)
```

```
my_name = "Luke"
```

Q: Why the error?

A: Just like you can't read a book before it's written, you can't ask Python to remember something that it never learned.

Variables pt 2

Variable names should start with a letter.

☒ `2cool = "Too cool"`

☑ `too_cool = "Too cool"`

Some names are built into Python and should not be used as variable names.

☒ `print = "Hi"`

☑ `message = "Hi"`

You can add one variable to another if they are both **strings** (text between quotes, e.g. `"Luke"`).

- Run:

```
my_name = "Luke"
sentence = " is rad."
print(my_name + sentence)
```

Variables pt 3

You can even add variables to strings.

- Run:

```
ben_says = "With great
power "
print(ben_says + "comes
great responsibility.")
```

You can change the value of a variable in your code.

- Run:

```
ben_says = "With great
power "
ben_says = "Uh oh, here "
print(ben_says + "comes
great responsibility.")
```

Variables pt 4

Q: Why did it only print out the second value?

A: If you use the assignment operator (=) on an existing variable, you overwrite it. We basically told Python to 'change its mind' in the example above.

You can set a variable equal to another variable.

- Run:

```
color = "blue"
eye_color = color
print(eye_color)
```

Variables pt 5

 **Activity:**

- Make your own "fake quote generator" based on the example below:

```
verb = "steal"

noun_plural = "trees"

famous_person = "Albert Einstein"

print("You should never " + verb + " "
+ noun_plural + ". Ever.")

print("- " + famous_person)
```

Outputs:

```
You should never steal trees. Ever.
- Albert Einstein
```

- Mix it up to create your own format. Get creative!
- Each variable is a "blank". Ask a friend to fill in the blanks, then change the variables accordingly.
- Read the hilarious output!

Save this code somewhere, as we'll use it again!

Hint: Don't show them your screen until the output arrives. This way it's a surprise.

Variables pt 6

Some Possible Answers:

```
huge_number = "900"

cheap_item = "paperclip"

famous_person = "King Henry VIII"

print("I once bought a " + cheap_item +
" for " + huge_number + " dollars. Best
money I ever spent.")

print("- " + famous_person)


animal_plural = "turtles"

print("I would not be the person I am
today had I not been raised by " +
animal_plural + ".")
```

Explanation:

Students should make their own version of this fill-in-the-blanks game to encourage them to use what they've learned in a creative way.