CIS 1100

Hello, World! (Lecture)

Python

Fall 2024

University of Pennsylvania

Special Characters

Our first program was

```
print("Hello World")
```

which prints `Hello World!`

Special Characters

Our first program was

```
print("Hello World")
```

which prints `Hello World!`

What if we wanted to print something like:

`My favourite album lately has been "MM... Food"`

Special Characters

Let's try this:

```
print("My favourite album lately has been "MM... Food"")
```

... this does not work :(

Escaping Special Characters

the `"` are used to mark the start and stop of a string, we need to tell the computer that we don't actually want to stop the string when it hits a `"` if we want to print it.

We can do this by "escaping" the `"` with a `\` e.g.:

```
print("My favourite album lately has been \"MM... Food\"")
```

the `\` will tell the computer to treat the next character "special".

Activities:

Which of these lines of code work:

- `print("hello\")` (M1) A -> True, B-> False
- `print("howdy\"\"\" partner\"")` (M2) A -> True, B-> False

Any guess for how to print the `\` character itself? `print("\")` does not work. (S7)

Calling Functions

in our program, we have already been making use of calling **functions**

`print()` is a function. Funtions are special bits of code that either:

- Modify something (e.g. printing to the terminal)
- Evaluate to something
- Both

You know you are calling a function when you see something like `function_name()` (Parenthesis on the end being key.)

print

The `print()` function prints something out so that we can see it. Withing the `()` we provide the string we want it to print.

```
# prints the string "Hello World!"
print("Hello World!")
```

len

the `len()` function calculates the length of something that we give it in the `()`.

For example `len("Hi")` is `2` since `"hi"` has two characters

`len()` doesn't do anything outside of getting the length, so this program ends up just printing hello:

```
len("hello")    # doesn't print anything, calculates the length then does nothing with it
print("hello")
```

Combining len and print

We can use both `len()` and `print()` together:

```
print(len("hello"))  # prints "5"
```

`len()` evaluates to the value `5` and then we put that into `print` to print it.

Note: doing `len(print("Hello"))` doesn't make sense since `print` doesn't evaluate to anything

input()

Input is a function that does two things: it both prints and evaluates to whatever the user typed in.

Consider:

```
print(input("What is your name?"))
```

This program prints out "What is your name", reads what you type in, and prints it back out.

Activity:

What does this program print: (S8)

```
print(len("How are you?"))
```

What does this program print: (S9)

```
len("Pink")
print("Python")
```

Describe what this program does: (L11)

```
print(len(input("Give me something: ")))
```

Functions with multiple inputs

Sometimes a function can take in multiple inputs. `print()` can do this.

When we do such a thing, we use a comma `,` so separate the inputs.

```
print("MM...", "FOOD")
```

Two strings here instead of one, with a comma to separate them.

This prints out together `"MM... FOOD"`

Notice how both strings get printed with a space between them

Explaining name = input()

In the pre-lecture video, we did something like:

```
name = input("what is your name?")
```

Lets dissect this a little bit

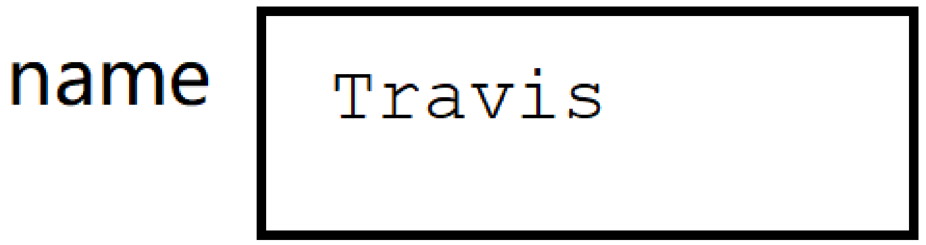
Intro to Variables

Lets start with a simpler example:

```
name = "Travis"
```

What this does is it creates a ***Variable*** stored named "name" holding the value `"Travis"`.

You can think of a variable sort of like a box with a name attached to it. The value in the box can change over the course of the program, but can only hold one thing at a time.

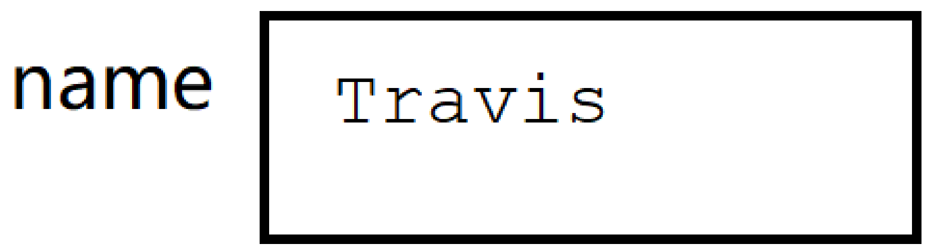


Printing Variables

We can print variables, similar to how we print strings:

```
name = "Travis"
print(name)  # prints Travis
```

When we give the name of the variabled, it tells python to see what is stored "inside that box"



name

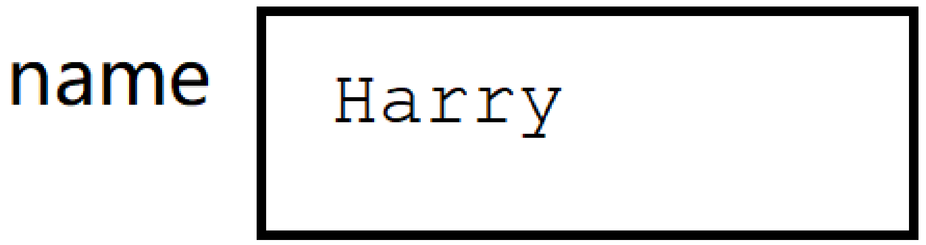
Travis

Variables Change over Time

Variables can change over the course of a program, and can only hold one value.

Consider:

```
name = "Harry"        <-----
name = "Travis"
print(name)
```

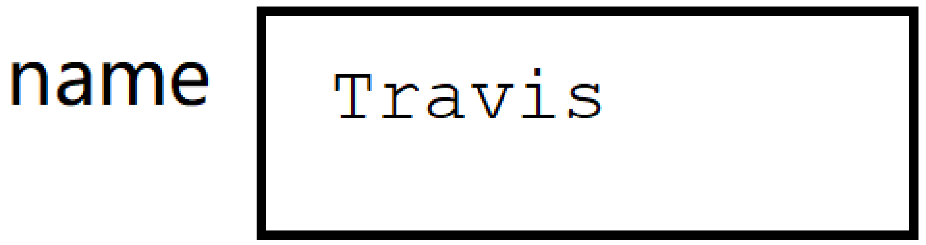


Variables Change over Time

Variables can change over the course of a program, and can only hold one value.

Consider:

```
name = "Harry"
name = "Travis"          <-----
print(name)
```



name

Travis

Explaining name = input

When we said:

```
name = input("What is your name?")
print("Hello!", name)
```

This will:

- ask you your name
- read what you typed in, and store it into a variable ("Box") called `name`
- prints out `"Hello!"` and the name that you typed in, which was stored in the variable `name`.

Activity:

What does this program do? (C12)

```
first_name = "Travis"
last_name = "McGaha"
print("Hi! My name is", first_name, "last_name")
```

Reminder:

- No lecture on Monday, next lecture on Wednesday 09/04
- There is another check-in due before that lecture as well.
- Volunteer Office Hours will start next week (keep an eye on Ed)
- No recitation next week, those start on 09/09 and 09/10