

CIS 11000

Hello, World! (Lecture)

Python

Spring 2025

University of Pennsylvania

Our First Program

Take a look at our very first program below!

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

We saw that this prints out "Hello World!" to the terminal.

Special Characters

Take a look at our very first program below!

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

We saw that this prints out "Hello World!" to the terminal.

But, what if we wanted to print our favrotie qoute from Harry Potter?

```
"Go boil yer head, both of yeh",  
said Hagrid. "Harry - yer a wizard".
```

Special Characters

Let's try it out:

```
print("""Go boil yer head, both of yeh", said Hagrid. "Harry - yer a wizard".")
```

Special Characters

Let's try it out:

```
print("""Go boil yer head, both of yeh", said Hagrid. "Harry - yer a wizard".")
```

So, that totally didn't work. What gives?

Escaping Special Characters

In Python, quotation marks " are used to mark the start and end of a string.

If we want to *actually* print a quotation mark, we need a special way to tell python that it is *a part of the string*, rather than indicating it's end.

Escaping Special Characters

In Python, quotation marks `"` are used to mark the start and end of a string.

If we want to *actually* print a quotation mark, we need a special way to tell python that it is *a part of the string*, rather than indicating it's end.

We can do this by *escaping* the `"` with a `\`.

As an example:

```
print("\Go boil yer head, both of yeh", said Hagrid. \Harry - yer a wizard\".")
```

`\` tells the computer to treat the next character as "special".

In other words, it allows it to be a part of the string itself! Neat.

Lecture Activities

Which of these lines of code work?

Handout: M1

- `print("hello, you're fantastic!\")`
 - A: True, it works.
 - B: False, it doesn't work.

Handout: M2

- `print("howdy\"\"\" partner\"")`
 - A: True
 - B: False

Handout: S7

How can we print the `\` character itself? *Hint: `print("\")` does not work.*

Calling Functions

We are one day in and we are making use of calling **functions!**

Our first function in CIS1100 is `print()`.

Functions are special portions of code that accomplish something for us.

As an example they might:

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

You can identify when a function is called when you see something like `function_name()`.

Parenthesis on the end being key. Without it, you are not calling a function.

print()

The `print()` function prints things out so that we can see it in the terminal.

Within the `()` we must provide the string we want it to print.

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

Reminder: # indicates the start of a comment in python

len()

The `len()` function calculates the **length** of the thing we pass in within the `()`.

For example, `len("Hi")` is `2` because `"hi"` has two characters. In other words, its length is 2 characters!

On its own `len()` doesn't do much. So this program ends up just printing hello:

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

len() and print()

To make `len()` useful, let's use it with `print()`:

```
print(len("Demure")) # prints "6"
```

`len()` *evaluates* to the value `6`, then we pass that into `print` to print it.

Note: `len(print("Demure"))` is incorrect because `print` evaluates to nothing.

input()

`input()` is a very useful function that does two things.

It both **prints** what you pass in and **evaluates** to what the user typed in.

Let's take a look at the following code:

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

First, this program prints out "What is your name".

Then, it reads what you type in (*after you hit enter*), and evaluates to that.

Lecture Activities

What do the following lines of code print?

Handout: S7

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

Handout: S8

```
print(len("Joel said, \"Lock in.\"))
```

Describe what this program does.

Handout: L11

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

Functions with Multiple Inputs

Some Functions can take in multiple inputs.

`print()` can take multiple inputs!

When we would like to pass in multiple inputs, we use a comma `,` to denote the next input.

Consider the following code:

```
# string 1: "\"Yer a wizard harry\"","  
# string 2: "said Hagrid"  
print("\"Yer a wizard harry\"","said Hagrid")
```

This prints out together `"Yer a wizard harry", said Hagrid`

Note: When passing in more than one string, they will be printed together. But, they will have 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 = "Joel"
```

This creates a *Variable* named "name" holding the value "joel".

You can think of a variable like a box with a name attached to it. Only, this boxes named is well, "name".

- the value in the box can change.
- the name of the box can not change.

name



Joel

Printing Variables

We can print variables similar to how we print strings.

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

When we pass in the name of the variable, this tells python to see what is stored "inside that box".

name



Joel

Variables Change over Time

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

Consider:

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

Reminder: Code runs from top to bottom!

name

Harry

Variables Change over Time

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

Consider:

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

Reminder: Code runs from top to bottom!

name

Joel

Explaining name = input

Consider the following:

```
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, [name] !"`, where `[name]` is what you typed in!

Lecture Activity

What does this program do? (C12)

```
first_name = input("First Name: ")
last_name = input("Last Name: ")

print("Hi! My name is", first_name, "last_name")
print("My first name is", len(first_name), "letters long.")
print(" ")
print("Good bye!")
```

Reminder:

- No lecture on Monday, next lecture on Wednesday, Jan 22nd.
- There is another check-in due before that lecture as well.
- Office Hours will start next week (keep an eye on Ed)
- No recitation next week, those start on 01/27 and 01/28.