Once you're seated, please respond to the poll at **pollev.com/compunc**

If you are not registered for PollEverywhere, please go ahead and do so before class begins!

Lecture 01

Take on Me

Practice: Primitive Data Types, Variables Introducing: `main` Function, Comments, Statements

Not enrolled in COMP110 yet? Please wait until after class for questions.

If you were not here Tuesday, you cannot get on the waitlist at this point.

If you were here Tuesday, check your e-mail for an update from me.

Not enrolled in COMP110 yet?

- If you were not here Tuesday:
  - **Bad news, unfortunately. Despite it appearing there are seats available on Connect Carolina, those open seats, plus 20 more seats still on our internal waitlist were claimed on Tuesday.**
  - **You are welcome to try COMP116 this Fall or COMP110 in the Spring.**

- If you were here Tuesday:
  - You should have received an e-mail update from me yesterday.

Videos 0 through 3 Graded Warm-up Questions

- You may have out a single sheet of handwritten notes
- Complete the 10 questions in order
  - Caution: PollEv does not allow going back to change responses
- We will spend 10 minutes on this. When complete, please do not do anything to distract your neighbors.
- Open **pollev.com/compunc**
  - Register quickly if needed.

Challenge Question #0 - pollev.com/compunc

- Solve for yourself with paper/pencil then talk with your neighbors to see if you came to the same answer.
  - Don't use an interactive programming REPL!

- What does the following expression evaluate to?

```
1 + 2 ** 3 % 3 + "4"
```

Primitive Data Types Highlights

- *Every **value***, or piece of data, has a specific ***type***

- Primitive Types (simplest data types)

  1. **boolean**
     - Only two possible values: true or false
     - We will explore its operators next week
  2. **number**
     - Integers: -1, 2, 3
     - Decimals: 3.14
     - With `number` type data you have arithmetic operators like `**, *, /, %, +, -`
  3. **string**
     - Textual data surrounded by Quotes
     - Example: "Hello, World"
     - Strings can be text that resembles a number: "3.14"
     - You can join a `string` with another `string/number/boolean` using concatenation +

REPL - Playing with Data Types

- In Google Chrome, navigate to: **bit.ly/repl-110**

- Open the REPL:
  - Windows: Control + Shift + J
  - Mac: Command + Option + J

- Let's try the previous expression as well as some variations of it:
  - **`1 + 2 ** 3 % 3 + "4"`**
  - **`1 + 2 ** 3 % 3 + 4`**
  - **`"1" + 2 ** 3 % 3 + "4"`**

Questions on Primitive Data Types or Operations?

Challenge Question #1 - pollev.com/compunc

- What is the value of the variable **x** after this code listing completes?

```
let x = 110;
let y = 101;
let temp = x;
x = y;
y = temp;
```

Challenge Question #2 - pollev.com/compunc

- What is the value of the variable **e** after this code listing completes?

```
let a = "1";
let b = "2";
let c = "3";
let d = b + c + b;
let e = a + d + a;
```

Variables Highlights

- Variables allow your programs to store, change, and lookup values in memory.
  - You will use variables for all sorts of purposes in programs, like storing information about a the person using your app, counting the number of times something happens, the results of computations, and more!

- **Declare** and **Initialize**:
  ```
  let <variableName> = <initial value>;
  ```

- **Assign** new values:
  ```
  <variableName> = <new value>;
  ```

- **Access** a variable's value by its name:
  ```
  <variableName>
  ```

REPL - Playing with Variables

- In Google Chrome, navigate to: **bit.ly/repl-110**
- Open the REPL:
  - Windows: Control + Shift + J
  - Mac: Command + Option + J
- Let's try the previous example:

```
let a = "1";
let b = "2";
let c = "3";
let d = b + c + b;
print(d);
let e = a + d + a;
print(e);
```

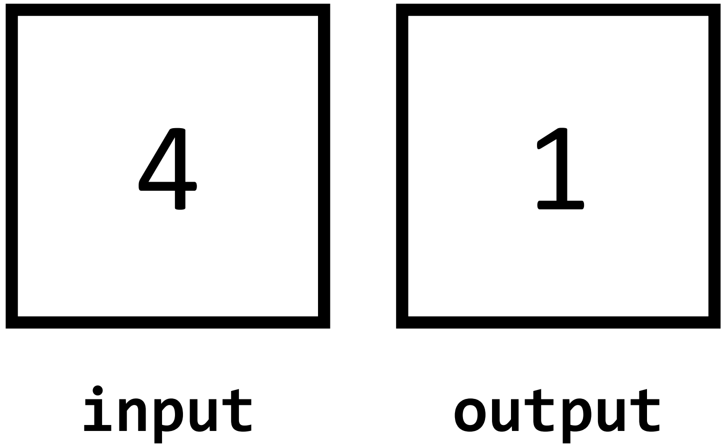
Questions on Variables?

CPU Hat Practice Puzzle

- On a piece of paper, make two boxes.
  - If you're writing in pen, make them big enough to scratch out their contents with room to write more.

- Label one "input", write 4 in it.

- Label the other "output", write 1 in it.



Instructions

1. If **input is at least 2**,
   then continue to **INSTRUCTION 2**,
   otherwise, **JUMP** to **INSTRUCTION 5**

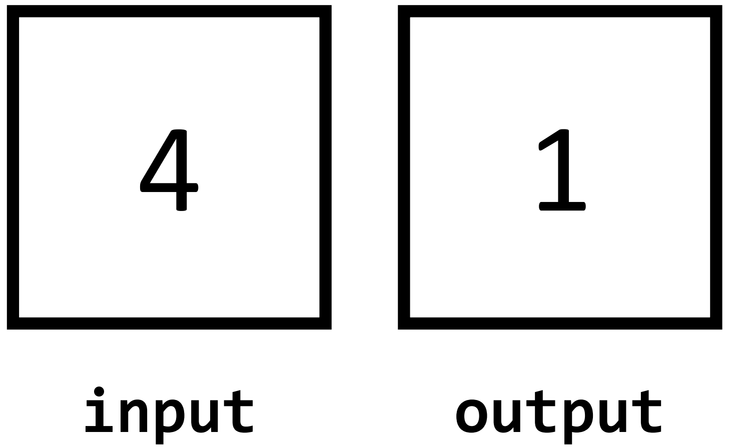
2. output = input * output;

3. input = input - 1;

4. JUMP to **INSTRUCTION 1**

5. The answer is stored in **output**.

6. EXIT



Instructions

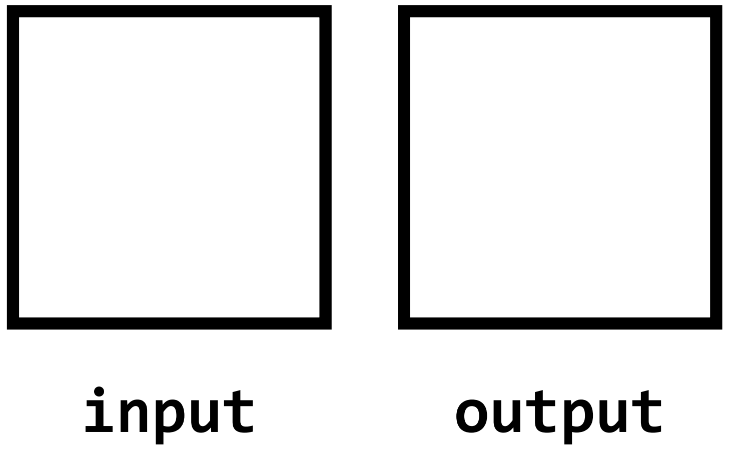
1. **IF input is at least 2**,
    THEN continue to **INSTRUCTION 2**,
    ELSE JUMP to **INSTRUCTION 5**

2. **output = input * output;**

3. **input = input - 1;**

4. **JUMP** to **INSTRUCTION 1**

5. The answer is stored in **output**.
6. **EXIT**



And now, let's write a stored program…

Your Software Probably Won't Work Today

- That's totally *OK and* ***Expected****!*
- If yours is not working, don't stress, just follow along with me!
- But *come to the Getting Started Open House tomorrow from 12p - 6p in Sitterson 008!*

// Code Comments (1/2)

- We can add notes for ourselves in code using **comments**
- The computer ignores comments when running your program
- Single line comments begin with a //

```
// This is a single line comment
```

- Multi-line comments are surrounded by an opening **/*** and closing ***/**

```
/* This
   is
   a multi-line comment
*/
```

- Comments are also useful for ignoring code you've written without deleting it.

// Code Comments (2/2)

- In the early days of programming, we recommend writing comments liberally in your code to explaining what your program is doing in English.

- Comments are *free* so use them liberally.

- Comments are the easiest way to "take notes" when we are working on examples in lecture.

Breaking Down "Hello World" (1 / 4)

- We begin with **importing** functionality from libraries
- Here we're importing a function named **`print`** from the `introcs` library
- The `introcs` library includes many other functionalities, too. We'll continue exploring it soon.

```
// Hello World Demo

import { print } from "introcs";

export let main = async () => {

    print("hello, world");

};

main();
```

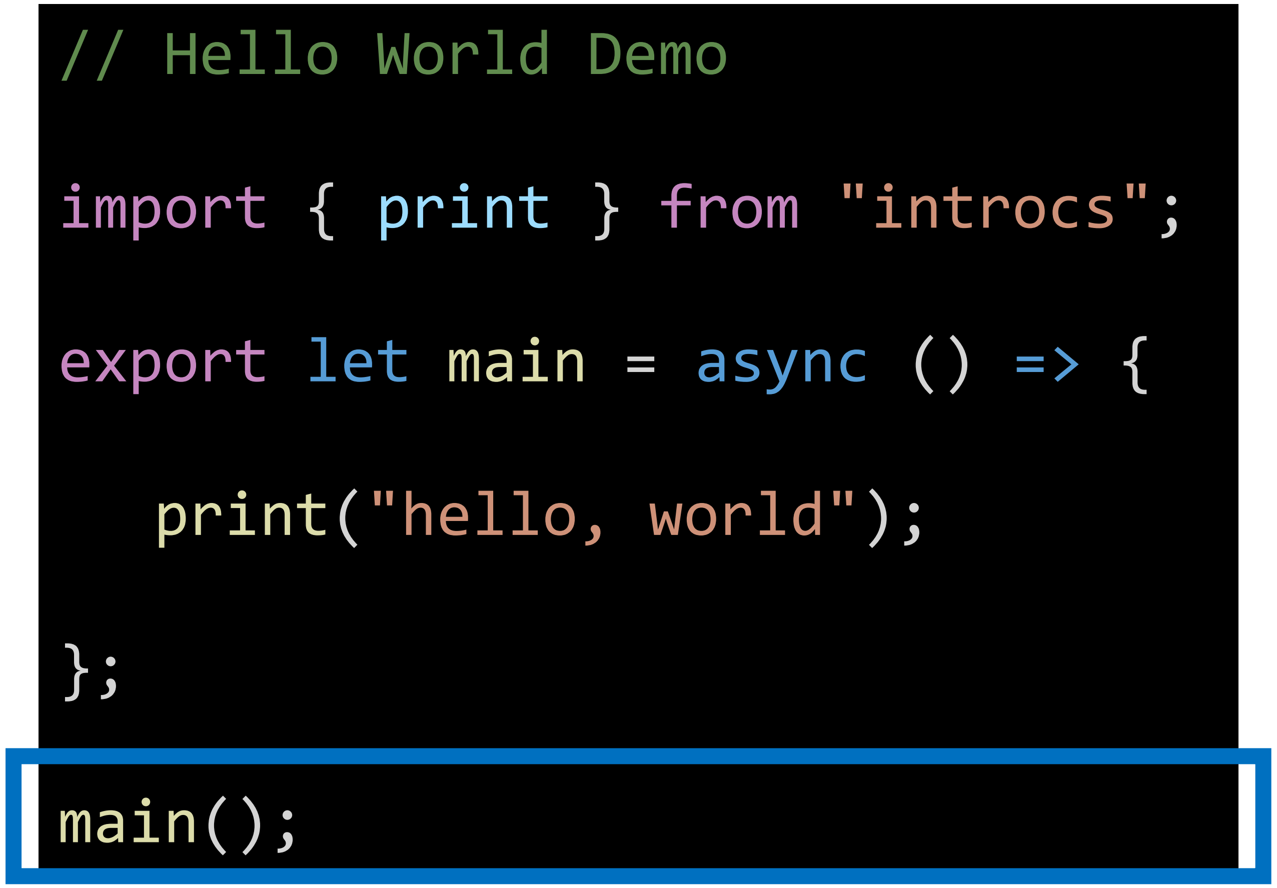
Breaking Down "Hello World" (2 / 4)

- The programs we write will have a `main` function defined in this way.
- This definition contains the instructions we want the program to follow (*print "hello, world"*)
- ***Soon*** you will learn exactly what each of these words and symbols means.
- For now, we will treat the `main` function's definition as magic.

```
// Hello World Demo

import { print } from "introcs";

export let main = async () => {

    print("hello, world");

};

main();
```

Breaking Down "Hello World" (3 / 4)

- The last line of code in our program **calls**, or invokes, the **`main`** function
- This line tells our main function to proceed so our program begins.
- You will see this in every program, as well.



```
// Hello World Demo

import { print } from "introcs";

export let main = async () => {

   print("hello, world");

};

main();
```

Breaking Down "Hello World" (4 / 4)

- Initially, you will write all of your code *inside* of the main function.
- These are the instructions your program will follow.
- Later, we will learn how to write our own functions that are defined outside of **main**.

```
// Hello World Demo

import { print } from "introcs";

export let main = async () => {

    print("hello, world");

};

main();
```

Statements (1/2)

- **Statements** are equivalent to an English sentence
- Statements *usually* end with a semi-colon ;
  - Like sentences end with a period!
- Each statement is an instruction you are giving to the computer.

```
print("hello, world");
```

That's a **statement**!

"Print 'hello, world' to the screen."

Statements (2/2)

- In a stored program, the computer will not carry out our instructions until we **run** the code
  - When you *write* programs, it is like you are writing a recipe down
  - When you *run* a programs, the computer is like the chef *following* your recipe
- Before the first statement runs, your program is a barren, empty world
  - *Your* code builds up its own little world piece-by-piece

```
print("hello, world");
```

That’s a **statement**!

“Print ‘hello, world’ to the screen.”

tfw you write your first stored program...

Quiz 0 - The First Quiz

- Tuesday - 8/28

- You cannot use notes on quizzes like you can on warmup questions

- Covers:
  - Important Course Policies to Know (expect true/false questions on this)
    - http://comp110.com/topics/getting-started/important-course-policies

  - Videos 2 and 3:
    - Primitive Data Types
    - Variables

- Don't forget your:
  - Pencil - we will provide the paper
  - ONECard

- If you are registered to take with Accessibility Resources & Service and prefer SASB
  - Come to start of lecture and take at SASB once quiz starts