BOOLEAN AND CONDITIONAL LOGIC

Objectives

- Learn how to get user input in Python
- Learn about "Truthiness"
- Learn how to use comparison operators to make a basic program

User Input

There is a built-in function in Python called "**input**" that will prompt the user and store the result to a variable.

```
name = input("Enter your name here: ")
Enter your name hereTyrion Lannister
name
'Tyrion Lannister'
```

BOOLEAN EXPRESSIONS

Conditional Statements

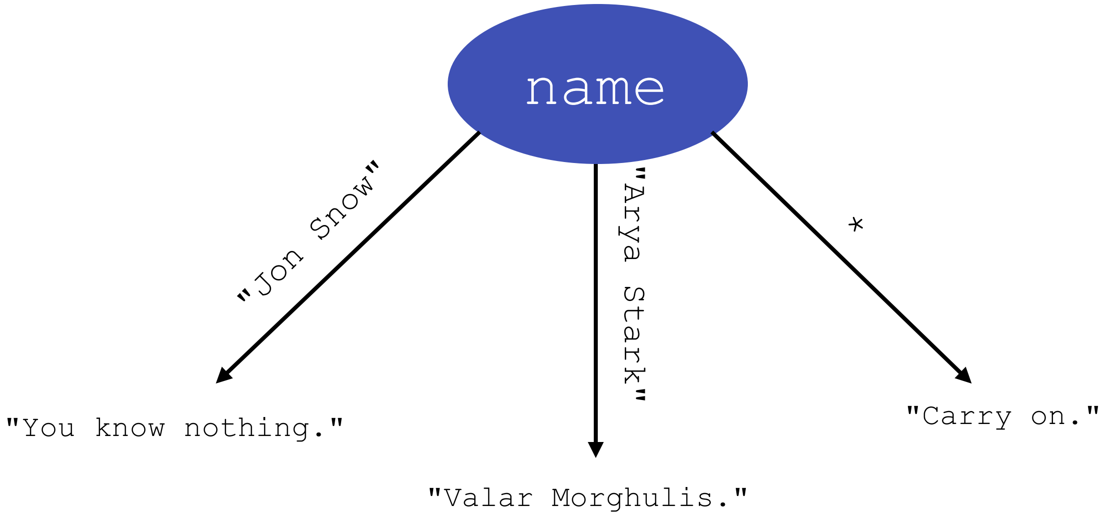
Conditional logic using *if* statements represents different paths a program can take based on some type of comparison of input.

```
if some condition is True:
do something
elif some other condition is True:
do something
else:
do something
```

Conditional Checks

Conditional logic using *if* statements represents different paths a program can take based on some type of comparison of input.

```
if name == "Arya Stark":
    print("Valar Morghulis")
elif name == "Jon Snow":
    print("You know nothing")
else:
    print("Carry on")
```



Truthiness

In Python, all conditional checks resolve to **True** or **False**.

```
x = 1
x is 1   # True
x is 0   # False
```

We can call values that will resolve to True "**truthy**", or values that will resolve to False "**falsy**".

Besides False conditional checks, other things that are naturally falsy include: empty objects, empty strings, *None*, and zero.

Comparison Operators

Here is a list of comparison operators.

In the examples, **a = 1** and **b = 1**

Op	**What it does**	**Example**
==	Truthy if **a** has the same value as **b**	a **==** b # True
!=	Truthy if **a** does **NOT** have the same value as **b**	a **!=** b # False
> <	Truthy if **a** is greater than **b** Truthy if **a** is less than be **b**	a **>** b # False a **<** b # False
>= <=	Truthy if **a** is greater than or equal to **b** Truthy if **a** is less than or equal to **b**	a **>=** b # True a **<=** b # True

Logical Operators

In Python, the following operators can be used to make Boolean Logic comparisons or statements:

Op	**What it does**	**Example**

and	Truthy if both **a** AND **b** are true
	(logical conjunction)
		if a **and** b:
		print(c)

or	Truthy if either **a** OR **b** are true
	(logical disjunction)
		if am_tired **or** is_bedtime:
		print("go to sleep")

not	Truthy if the opposite of **a** is true
	(logical negation)
		if not is_weekend:
		print("go to work")

is vs. "=="

"It depends upon what the meaning of the word '*is*' is."
- Bill Clinton

In python, "**==**" and "**is**" are very similar comparators, however they are not the

```
a = 1
a == 1   # True
a is 1   # True
```

```
a = [1, 2, 3]  # a list of numbers
b = [1, 2, 3]
a == b   # True
a is b   # False
```

```
c = b
b is c   # True
```

"**is**" is only truthy if the variables reference the same item *in memory*

BOUNCER CODE-ALONG

(with a nested conditional)

Bouncer Solution 1

The following program, **bouncer.py**, determines whether the user can enter the club or not.

```
age = input("How old are you: ")
if age:
        age = int(age)
        if age >= 18 and age < 21:
                print("You can enter, but need a wristband!")
        elif age >= 21:
            print("You are good to enter and can drink!")
        else:
                print("You can't come in, little one! :(")
else:
        print("Please enter an age!")
```

It also checks to make sure the user entered an age

Bouncer Solution 2

The following program, **bouncer.py**, determines whether the user can enter the club or not.

```
age = input("How old are you: ")
if age:
    age = int(age)
    if age >= 21:
      print("You are good to enter and can drink!")
    elif age >= 18:
      print("You can enter, but need a wristband!")
    else:
        print("You can't come in, little one! :(")
else:
    print("Please enter an age!")
```

Slightly refactored conditional logic

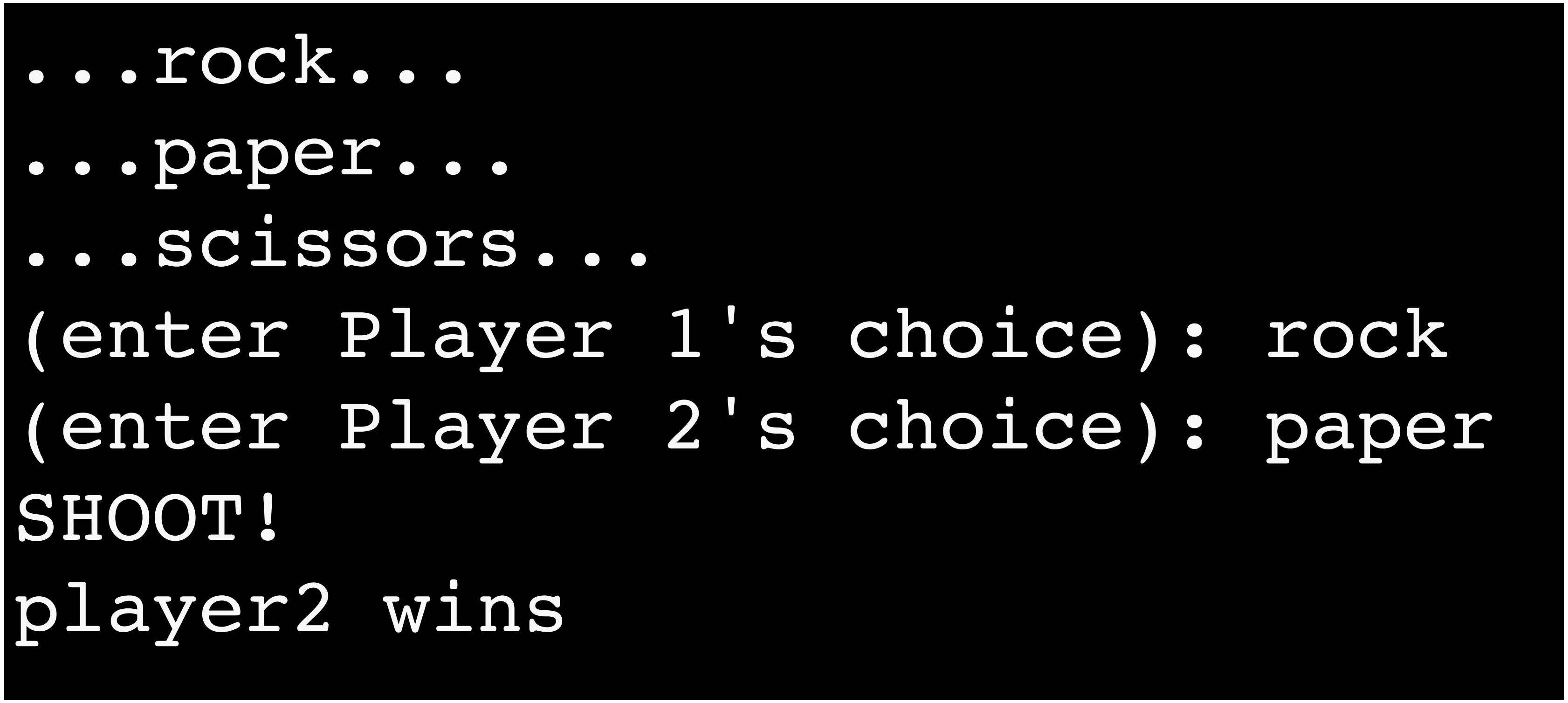
Recap

- Conditional logic can control the flow of a program
- We can use comparison and logical operators to make conditional *if* statements
- Conditional logic evaluates whether statements are truthy or not

ROCK 🤜
PAPER 🖐
SCISSORS ✌

"Try this on your own or treat it as a code-along"

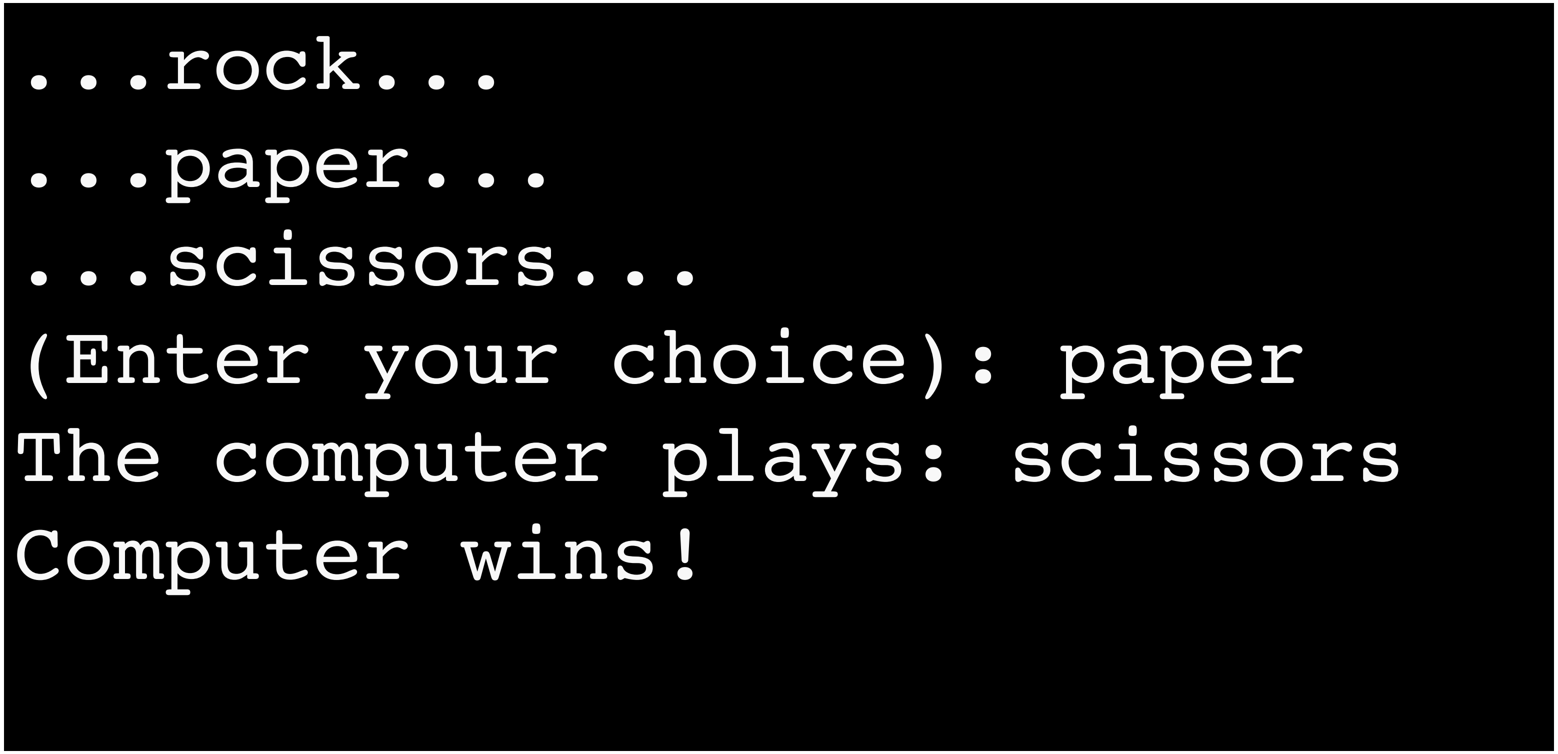
THE "BASIC" VERSION



```
...rock...
...paper...
...scissors...
(enter Player 1's choice): rock
(enter Player 2's choice): paper
SHOOT!
player2 wins
```

It may not be the most fun game ever made

THE "LESS BASIC" VERSION



```
...rock...
...paper...
...scissors...
(Enter your choice): paper
The computer plays: scissors
Computer wins!
```

Featuring the dumbest "AI" ever

HINT!

Research

You will need to import something

randint

🙀

YOUR
TURN