User! Input!

Programmer vs. User

```
1  print("Hello, wor
```

programmer

user

Why User Input?

So far, we've been writing programs that only interact with the user in 1 direction - displaying information out to them.

Today, we're gonna discuss how to add the other part - having the user interact with the program!

2 Necessary Elements

We need 2 things when we're getting input from the user:

- A way to get whatever the user types
- A way to store it

Using the input() function

When we use `input()`, we are prompting the user for information.

Similar to `print()`, we can put a message to the user inside of `input()`! This will let them know **what kind of information** they should be typing in.

Example:

```
input("happy?")
```

Is **WAY** worse than

```
input("Are you happy? (Yes/No) ")
```

because the second option tells the user what exactly they should input to have the program work correctly.

Saving Into Variables

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

When we use `input()`, our program pauses until the user presses **Enter** on their keyboard. They can choose to type things before pressing Enter, but they don't have to.

Type of User Input

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

What type should we expect to see here?

Type of User Input

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

What type should we expect to see here?

```
<type 'str'>
```

Type of User Input

```
age = input("What is your age?")
print(type(age))
```

What type should we expect to see here?

Type of User Input

```
age = input("What is your age?")
print(type(age))
```

What type should we expect to see here?

```
<type 'str'>
```

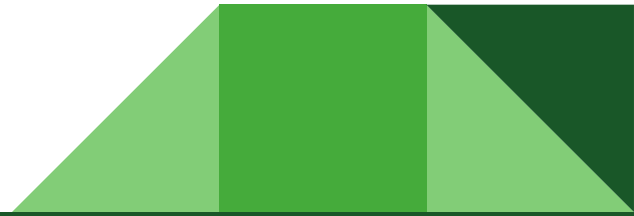
Typecasting

Typecasting is when we change a value's **type**.

We do this using the functions named after types - int(), str(), and float().

Each of these functions takes one piece of information, the value to be converted, and will hand back a converted value, so we need to do something with their output.

Usually we'll save it into a variable.



Typecasting Example

```
num = 42                    # Value = 42
word = str(num)             # Value = "42"
decimal = float(word)       # Value = 42.0
num_2 = int(decimal)        # Value = 42
```

Function Composition

That's a fancy word for when a function call is a parameter inside another function call.

e.g. `user_value = int( input( "BLAH BLAH" ) )`

Function calls are **always** evaluated from the *inside out*. Let's assume the user inputs 12.

```
user_value = int( "12" )
user_value = 12
```