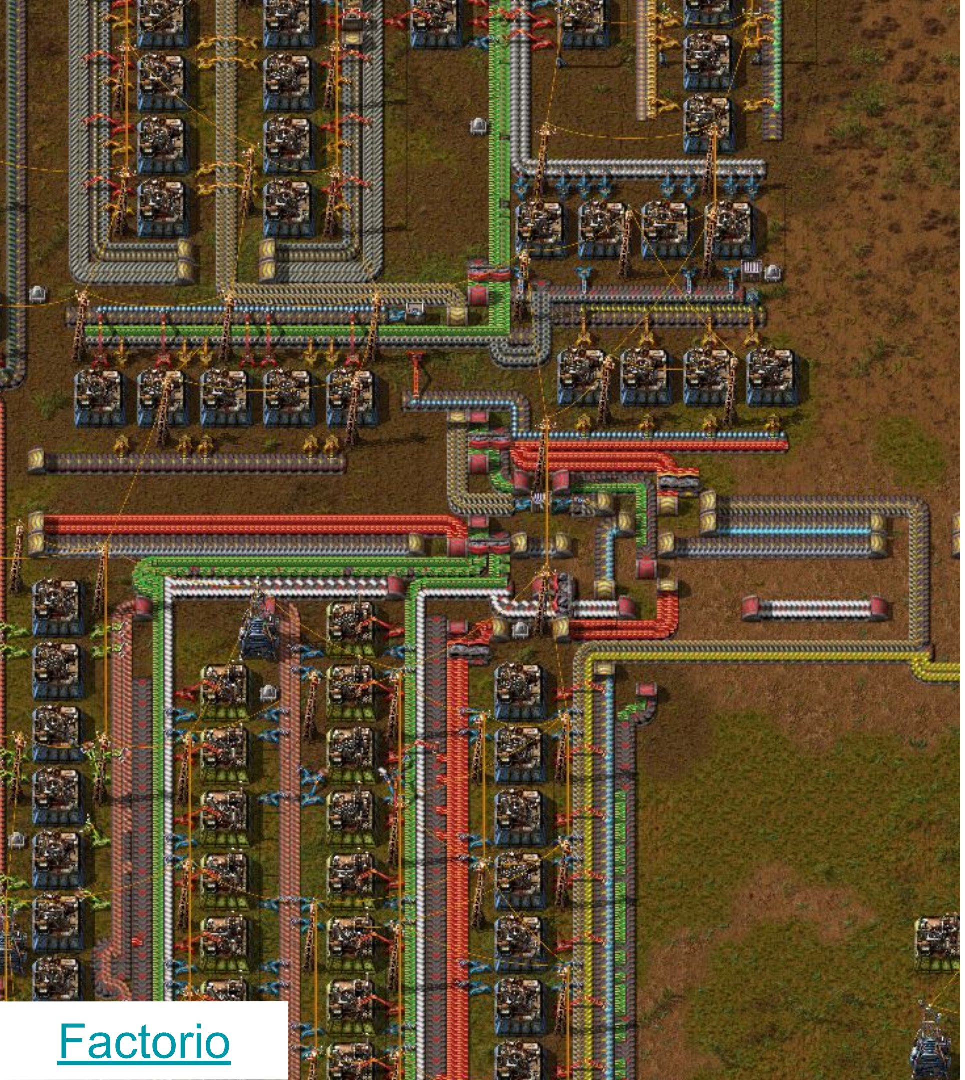

Factorio

Using basic functions

Quick review!

- What discussed three “parts” of a variable. What are they?
  - Its name, value, and type
- What data types have we discussed so far? What do they represent?
  - Ints (whole numbers), floats (numbers w/ decimals), strings (text)
- What are the values and types of these variables?

```
mystery_num = 16 % (2 + 1 * 3)
second_num = 5**2 / 2
```

Quick review!

- What discussed three “parts” of a variable. What are they?
  - Its name, value, and type
- What data types have we discussed so far? What do they represent?
  - Ints (whole numbers), floats (numbers w/ decimals), strings (text)
- What are the values and types of these variables?

  mystery_num = 16 % (2 + 1 * 3) —-> 16 % 5 = 1

  second_num = 5**2 / 2 —---------> 25 / 2 = 12.5

Practicing computational thinking

COLLEGE
COMMUNITY
COMPUTING
FUN
GRADES
GVSU
LAKER
LOUIE

L	X	F	U	N	S	E	D	A	R	G	G
N	O	D	R	Y	V	X	N	F	H	J	V
L	N	U	R	C	O	L	L	E	G	E	S
A	G	N	I	T	U	P	M	O	C	A	U
K	S	B	N	E	L	H	Y	A	L	Z	W
E	D	F	K	X	S	W	W	N	D	P	T
R	E	X	Z	I	U	V	I	F	N	K	U
C	O	M	M	U	N	I	T	Y	X	N	B

Practicing computational thinking

COLLEGE
COMMUNITY
COMPUTING
FUN
GRADES
GVSU
LAKER
LOUIE

L	X	F	U	N	S	E	D	A	R	G	G
N	O	D	R	Y	V	X	N	F	H	J	V
L	N	U	R	C	O	L	L	E	G	E	S
A	G	N	I	T	U	P	M	O	C	A	U
K	S	B	N	E	L	H	Y	A	L	Z	W
E	D	F	K	X	S	W	W	N	D	P	T
R	E	X	Z	I	U	V	I	F	N	K	U
C	O	M	M	U	N	I	T	Y	X	N	B

Practicing computational thinking - Activity

Functions

Functions are reusable chunks of code

You don’t have to copy-paste the chunk, just *call* the function!

Have we seen any functions so far in class?

Yes! print() and input() are both functions!

Built-in functions

Both print() and input() are built-in functions

They are ALWAYS available in Python, without doing anything extra

There are ~70 built-in functions:
https://docs.python.org/3/library/functions.html

Looking back at an old example:

```
print("Type something and then press enter!")
x = input()
print("x is:")
print(x)

print("Hit enter one more time...")
input()
y = x * 3
print("y is:")
print(y)
```

`input( )`

Waits for the user to write something in the terminal and then press enter

Returns that value as a string.

Why?

Strings can “hold” numbers, numbers can’t “hold” strings

So strings are more flexible, they are safer!

```
print(arg)
```

`print(arg)`

Some functions allow us to pass information to the function!

`print(arg)`

Some functions allow us to pass information to the function!

We do this via arguments:

`print(arg)`

Some functions allow us to pass information to the function!

We do this via arguments:

$$\underbrace{\texttt{print}}_{\text{function}}\texttt{(}\underbrace{\texttt{“this is a test”}}_{\text{argument}}\texttt{)}$$

`print(arg)`

Some functions allow us to pass information to the function!

We do this via arguments:

$$\underbrace{\texttt{print}}_{\text{function}}\texttt{(}\underbrace{\texttt{“this is a test”}}_{\text{argument}}\texttt{)}$$

Arguments can be variables or literals

`print(arg)`

Some functions allow us to pass information to the function!

We do this via arguments:

$$\underbrace{\texttt{print}}_{\text{function}}\texttt{(}\underbrace{\texttt{“this is a test”}}_{\text{argument}}\texttt{)}$$

Arguments can be variables or literals

`print` doesn’t return anything (i.e., returns None)

Other built in functions

What do you think these functions do / return?

```
max(2, 6, 1, 3, 4)

min(2, 6, 1, 3, 4)

abs(-3.14)

len(“dr. ferg”)
```

Other built in functions

What do you think these functions do / return?

`max(2, 6, 1, 3, 4) ->` Returns 6 (maximum argument)

```
min(2, 6, 1, 3, 4)

abs(-3.14)

len(“dr. ferg”)
```

Other built in functions

What do you think these functions do / return?

`max(2, 6, 1, 3, 4) ->` Returns 6 (maximum argument)

`min(2, 6, 1, 3, 4) ->` Returns 1 (minimum argument)

`abs(-3.14)`

`len(“dr. ferg”)`

Other built in functions

What do you think these functions do / return?

```
max(2, 6, 1, 3, 4) ->  Returns 6 (maximum argument)

min(2, 6, 1, 3, 4) ->  Returns 1 (minimum argument)

abs(-3.14)         ->  Returns 3.14 (absolute value)

len(“dr. ferg”)
```

Other built in functions

What do you think these functions do / return?

```
max(2, 6, 1, 3, 4) ->  Returns 6 (maximum argument)

min(2, 6, 1, 3, 4) ->  Returns 1 (minimum argument)

abs(-3.14)         ->  Returns 3.14 (absolute value)

len(“dr. ferg”)    ->  Returns 8 (number of characters (length))
```

Correcting our past mistakes

```
print("Type something and then press enter!")
x = input()
print("x is:")
print(x)

print("Hit enter one more time...")
input()
y = x * 3
print("y is:")
print(y)
```

What's the problem here?

Correcting our past mistakes

```
print("Type something and then press enter!")
x = input()
print("x is:")
print(x)

print("Hit enter one more time...")
input()
y = x * 3
print("y is:")
print(y)
```

What’s the problem here?

x is a string!

Correcting our past mistakes

```
print("Type something and then press enter!")
x = input()
print("x is:")
print(x)

print("Hit enter one more time...")
input()
y = x * 3
print("y is:")
print(y)
```

What's the problem here?

x is a string!

How do we fix it?

Correcting our past mistakes

```
print("Type something and then press enter!")
x = input()
print("x is:")
print(x)

print("Hit enter one more time...")
input()
y = x * 3
print("y is:")
print(y)
```

What’s the problem here?

x is a string!

How do we fix it?

Convert it to a number!

Type casting / type conversions in Python

Converting between data types is easy!

Type casting / type conversions in Python

Converting between data types is easy!

Use `str()`, `int()`, or `float()`!

Type casting / type conversions in Python

What do you expect the outputs to be?

```
int(“12”)
float(“13.5”)
str(62)
str(9000.1)
```

Type casting / type conversions in Python

What do you expect the outputs to be?

```
int(“12”)     -> 12
float(“13.5”)-> 13.5
str(62)       -> “62”
str(9000.1)   -> “9000.1”
```

Correcting our past mistakes

```
print("Type something and then press enter!")
x = input()
print("x is:")
print(x)

print("Hit enter one more time...")
input()
y = x * 3
print("y is:")
print(y)
```

So how can we fix this code?

Correcting our past mistakes

```
print("Type something and then press enter!")
tmp = input()
x = float(tmp)
print("x is:")
print(x)

print("Hit enter one more time...")
input()
y = x * 3
print("y is:")
print(y)
```

So how can we fix this code?

Correcting our past mistakes (alternative)

```
print("Type something and then press enter!")
x = float(input())
print("x is:")
print(x)

print("Hit enter one more time...")
input()
y = x * 3
print("y is:")
print(y)
```

So how can we fix this code?

Type casting / type conversions in Python

What do you expect the outputs to be?

```
int(13.2)
int(13.9)
str(“this is a string”)
int(“13.2”)
float(“4x”)
```

Type casting / type conversions in Python

What do you expect the outputs to be?

```
int(13.2)
int(13.9)
str(“this is a string”)
int(“13.2”)
float(“4x”)
```

Try it!! I’ll always encourage you to test things out yourself! :^)

User-defined functions

Soon we’ll talk about writing our own functions!

Finding more functions

Python has a wealthy ecosystem of code

If you want a common function, *someone has likely already coded it!*

We can access extra code by importing modules

Example:

```
import math
x = math.sqrt(49)
```

Reading documentation

Documentation exists for both built-in functions:
https://docs.python.org/3/library/functions.html

And other modules:

https://docs.python.org/3/library/math.html

I ***strongly*** encourage you to get comfortable reading documentation!

Practice questions: How can you call print but avoid ending the line?
What does passing an argument to input do?