

4: Variables, functions, sys arguments, A1

Tuesday, January 11, 2022 10:13 PM

Announcements:

Don't forget quiz:

Event schedule

Variables

Tuesday, January 11, 2022 10:20 PM

Variable 

a named space to store data

use descriptive names

use `=` to put data in

```
1 my_quiz_grade = 8 #pronounce gets
2 print(my_quiz_grade)|
3
4 my_quiz_grad = 9 #secret typo!
5 print(my_quiz_grade)
6 #what will happen with this runs?
```

good Variable names

userWish

user_wish

bad

input 1
foo

(Ask
for
ideas) →

CamelCase

Snake_case

} Mostly personal preference
Python mostly uses
Snake.
Pick 1; stick to it!

Variables can be used anywhere

Variables can be used anywhere
a value is used

```
1 my_quiz_grade = 8
2 print(10)
3 print(my_quiz_grade)
4 print("my_quiz_grade") #what will print?
```

Variable Naming Rules

Must start with letter or underscore _
only letter or digits allowed
Case sensitive (name ≠ Name)
Try one and see what happens!

```
1 4thGrade = 4
2 |
```

Shell x

%%run helloWorld.py

Traceback (most recent call last):

[File "C:\Users\hardinc3\CSCI\Downlo
rld.py", line 1](#)

4thGrade = 4
^

SyntaxError: invalid syntax

Python language reserved some keywords
Google

Google



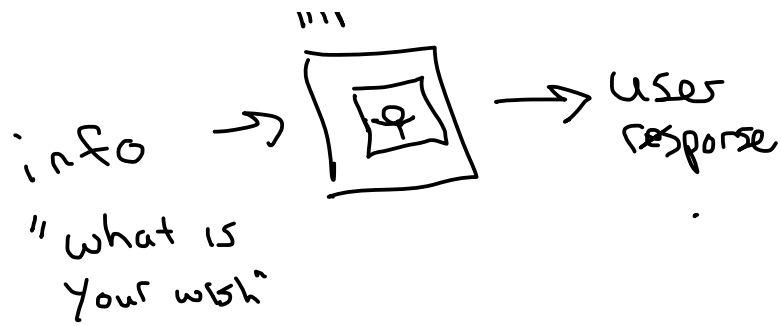
python reserved keywords

<https://docs.python.org> › [ref](#) › keywords ⋮

<https://docs.python.org/2.5/ref/keywords.html>

rennapi

```
1 global = 5 #see the color
2 global_count| = 5
```

could be
more than 1!

only
1 thing

Save with
= or
you just
dropped it



↓

Print("Your wish for", user_wish, "was granted")

Functions and Types

Sunday, January 9, 2022

3:33 PM

You want to change

string "1" to 1?

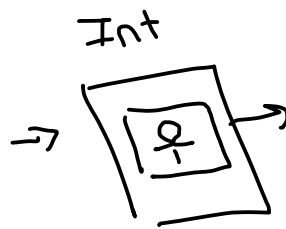
Python has a function for that!

int()

float()

str()

make this an
int pls 4.8



here you
go
4

```
2 test_type = int(4.8)
3 print(test_type)
4 #try for a couple different values in int
5
6 #second: you can have a function hand it's result
7 #directly to a different function!
8 print(int(4.8))
9 #inside most function 1st, then next, then next
10
11 print(str(4))
12
13 print(float(4))
14
15 print(float("4"))
16
17 print(int("4.01"))
```

```
1 #what if we want to do 3 things:
2 #string to float
3 #float to int
4 #print result
5 #try it!
6
7 demo = float("4.01")
8 demo = int(demo) #yes you can do this
9 print(demo)
10 #it does the work on the right first, then saves
11
12 print(int(float("4.01")))|
```

Don't forget Quiz!

System Arguments

Tuesday, January 11, 2022 11:39 PM

Print() \Rightarrow None
a keyword meaning No Value
input() \Rightarrow a string

```
1 a = input("Enter a number")
2 #what type does this come in as?
3 #what if we want it to be a float?
4 user_input1 = float(a)
5
6 #make it one line of code
7 user_input2 = float(input("Enter a number"))
```

But if you're testing code a bunch, it gets tedious to have to keep typing in the input. Luckily there's a way to automatically set some input at the time you run your code!

I'm going to show you while making the most common mistakes so you can see

helloWorld.py x

```
1 a = sys.argv[1]
```

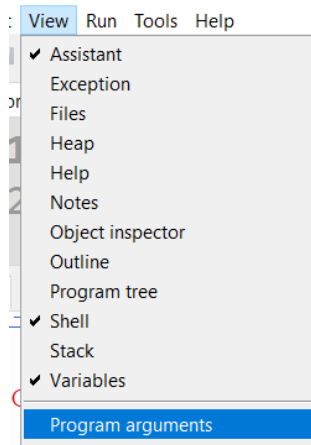
Shell x

```
a = sys.argv[1]
NameError: name 'sys' is not defined
```

```
1 import sys
2 a = sys.argv[1]
```

Shell x

```
import sys
a = sys.argv[1]
IndexError: list index out of range
```



Type here spaces between

No Comma

Program arguments: "put here", "whatever values", "you want to give your program", 5

```
helloWorld.py x
1 import sys
2
3 a = sys.argv[1]
4 b = sys.argv[2]
5 c = sys.argv[3]
6 d = sys.argv[4]
```

grab in numbered order

Name	Value
a	'put here,'
b	'whatever values,'
c	'you want to give your p
d	'5'
sys	<module 'sys' (built-in)>

comes in as string

```
Shell x
>>>
>>> %Run helloWorld.py 'put here,' 'whatever values,' 'you
    want to give your program,' 5
>>>
```

see it here also

Program arguments: 2, 3

```
World.py x
1 import sys
2
3 a = sys.argv[1]
4 b = sys.argv[2]
5 c = 2 + a
```

TypeError: unsupported operand type(s) for +: 'int' and 'str'

Modulus

Monday, September 26, 2022

2:50 PM

% modulus
keep only remainder

$7 \% 2$ is 3 with remainder 1

Very useful in CS
will be on Assignment #1!

(()) evaluate inside most parens 1st

You can evaluate expressions
in the shell

```
>>> 5%2
```

```
1
```

Assignment 1 preview

Friday, January 14, 2022 12:03 PM

Run the code a couple of times and show what the output looks like

Helpful code is now posted in the writeups folder