PYTHON CLASS 14

COMMON ERRORS AND LOOPS

HOW WAS SUMMER VACATION?

REVIEW CHECK

- Do you remember the rules about variable and function names?
- the basic types of variables?
- how to use if?
- how to make a function?
- how to use a function?
- how to make a list and use its functions?

SPELLING

- Name error

```
n = “Tom”

print(na)
```

```
if x.isupper():
        print(“It’s upper! Yay!”)
```

- Capitalization

```
import datetime

date = datetime.date(2022,6,21)
print(Date.weekday())
```

INDENTATION

- Your indentation doesn’t match the rest of the code.

```
import datetime

    date = datetime.date(2022,6,21)
  print(Date.weekday())
```

- Your indentation doesn’t match what you want to do.

```
import datetime
date = datetime.date(2022,6,21)
if Date.weekday() == 1:
        print(“It’s Tuesday!”)

print(“It’s a weekday”)
```

NUMBERS

- Zero division error
  - you are trying to divide a number by 0. Check where it happens.

```
b = 0
a = 7 / b
```

- Index error
  - you tried to use an index that is larger (smaller) than the list

```
li = [“Arthur”, “Lancelot”, “Gawain”, “Kay” ]
print(li[4])
print(li[-5])
```

TYPE/ASSIGNMENT

- Type error

```
import datetime
s = “Tuesday”
d = datetime.date(2022,6,22)
x = d + s
```

- Using assignment instead of equality
  - = versus ==

```
li = [“Arthur”, “Lancelot”, “Gawain”, “Kay” ]
li2 = [“Arthur”, “Lancelot”, “Gawain”, “Kay” ]

li = li2
print(li == li2)
```

FUNCTION RESULTS

- Remember, some functions give a return value, but some give None.
- You need to know which is which.

```
l = [“Hayao Miyazaki”, “Kentarou Miura”, “Takehiko Inoue”]

l = l.append(“Kohei Horikoshi”)
```

append doesn’t have a return value, so now l is lost!!!

FUNCTION ARGUMENTS

- Python always expects a function’s arguments to be in the same order.

```
name =  “Kentarou Miura”

name = name.replace(“Taro”, “Kentarou”) # name.replace(“taro”,
“Kentarou”) → What will the value of name be?

print(name) # name is still Kentarou Miura!!
```

this will not give an error, but the replace function expects the part we want to find **first**, **then** what we want to change it to

GLOBAL VS. LOCAL / ATTRIBUTE

- Unbound local error
  - You tried to assign a value to a global variable inside a function

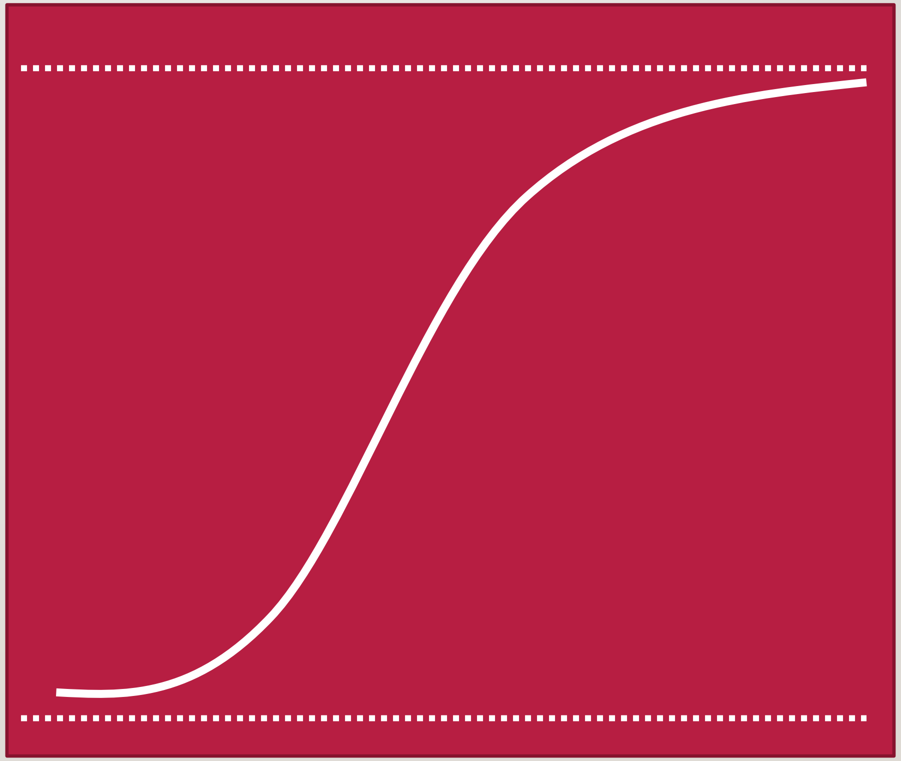
```
x = 5
def sq():
        x = x**2
```

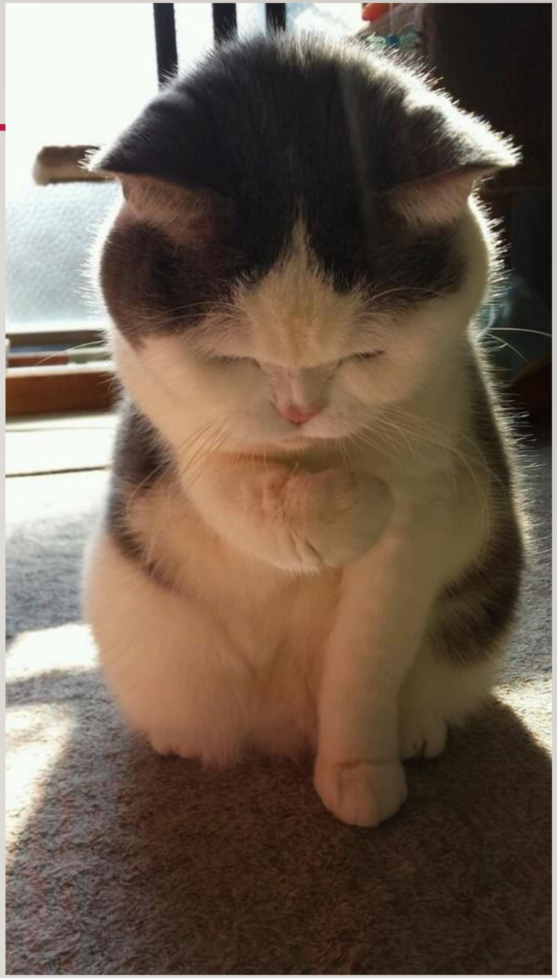
- Attribute error
  - you tried to access a function or piece of information that doesn’t exist

```
li = [“Arthur”, “Lancelot”, “Gawain”, “Kay” ]
li.isupper() #only exists for str variables
li.ascii_uppercase # only exists for the special string library, which you need to import
```

AND MANY MANY MANY MANY MORE

- As you get more experience as a programmer, you will encounter lots of other bugs and errors while you program.
- Also, you will become faster at finding them.
- You will never write perfect code, but that is OK!
- In life, we are all on a journey and we can try to get better as we go.





LOOPS AND LOOPING

REPETITION

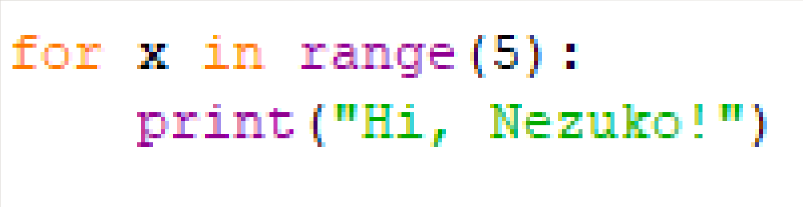
- https://youtu.be/KbiSxunJatM?t=34

GROUNDHOG DAY

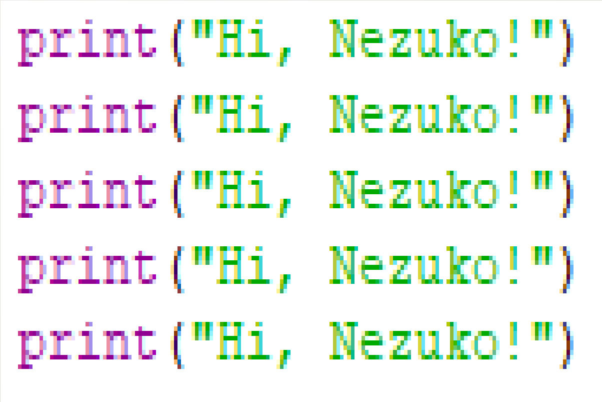
- Groundhog Day (Clip 3) - Repeated Dying Sequence - YouTube

LOOPS

- A loop is a way to repeat a statement or command.

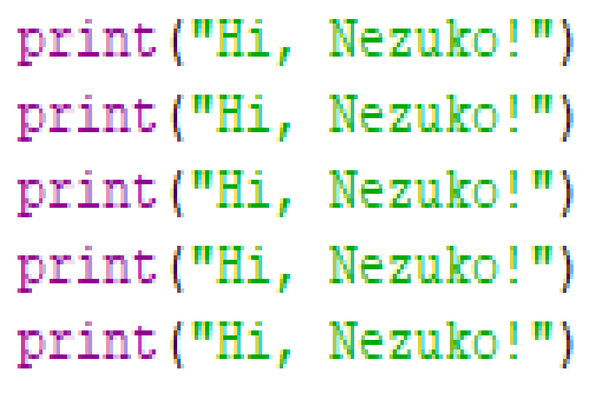


```
for x in range(5):
    print("Hi, Nezuko!")
```

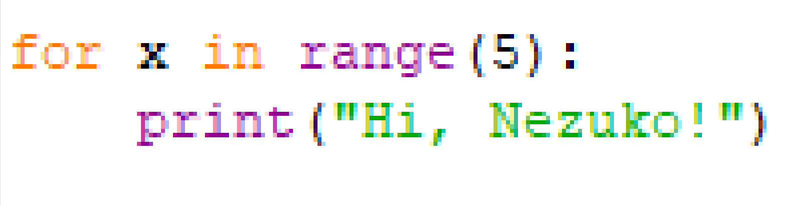


```
print("Hi, Nezuko!")
print("Hi, Nezuko!")
print("Hi, Nezuko!")
print("Hi, Nezuko!")
print("Hi, Nezuko!")
```

LOOPS



```
print("Hi, Nezuko!")
print("Hi, Nezuko!")
print("Hi, Nezuko!")
print("Hi, Nezuko!")
print("Hi, Nezuko!")
```

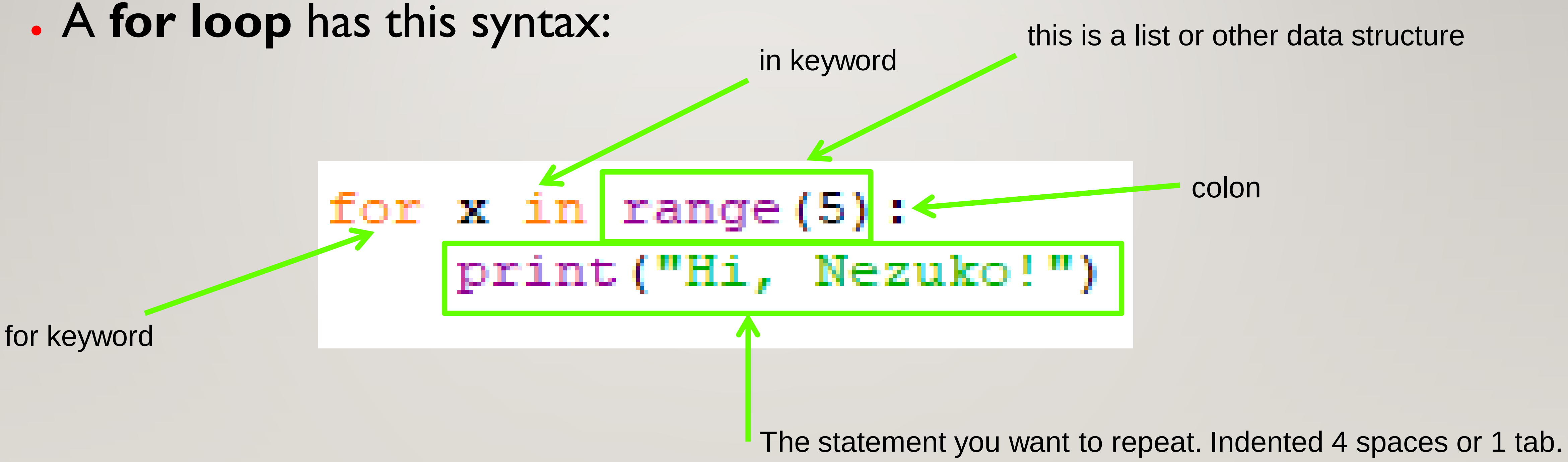


```
for x in range(5):
    print("Hi, Nezuko!")
```

What are loops good for?

LOOPS, CONTINUED

- In Python, the most basic kind of loop is a **for loop.**
- A **for loop** has this syntax:



```
for x in range(5):
    print("Hi, Nezuko!")
```

in keyword

this is a list or other data structure

colon

for keyword

The statement you want to repeat. Indented 4 spaces or 1 tab.

SPACES VERSUS TABS

- https://www.youtube.com/watch?v=SsoOG6ZeyUI

LOOPS AND LISTS

```
days = [“Mon”, “Tues”, “Wed” , “Thurs” , “Fri”, “Sat”, “Sun”]

for x in days:
        print(x)
```

What is the type of days?

What is the type of the information in days (What is the type of x?)?

WAYS TO CONTROL A FOR-LOOP

- Use range(a)
- Use range(a,b)
- Use in + a list (or dictionary, or other data structure)

VENDING MACHINE, AGAIN

- Let's go back to the vending machine project and use loops to show the product names and prices.

LOOPS, IF AND SCOPE

- Loops and **if** work very differently than functions.

```
def my_func(a):
        x = 5 * (a*a) + 11

my_func(10)

print(x) # this gives an error
```

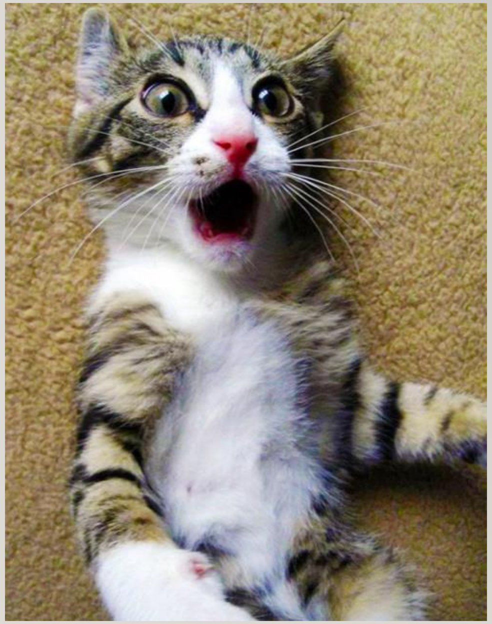
```
for x in range(10):
    a = 2 * x
    print(a)

print(a) #this is fine!
print(x) #this is fine, too!
```

```
n = "y"

if n == "w":
    b = "y2"
else:
    c = "x"

print(b) #this gives an error
print(c) #this is fine!
```



LOOPS AND SCOPE

- If you need to keep a value while running a for-loop, you should make the variable before it.

```
for x in range(10):
	a = 0
	a = a + x

print(a)
```

a is set to 0 each time you run through the loop, so the last value is 9…

```
a = 0

for x in range(10):
	a = a + x

print(a)
```

a is set to 0 outside the loop, so previous value gets added to it each time

LOOP EXERCISES

- Create a loop that counts from 0 to 100
- Create a loop that multiplies the numbers from 1-20.
- Create a loop that adds random numbers to a list.
- Use a loop to find the largest and smallest numbers, and the average.

WAYS TO CONTROL ACTION INSIDE A LOOP

- You can use if-statements and break or continue to control action

```
li = [2,4,6,8,10,11,14,16,18]

for x in li:
    if x % 2 != 0:
        break
    print(x)
```

we want to stop the loop completely if we have an odd number

```
li = [1,4,9,16,25,0,36,49,64]

for x in li:
    if x == 0:
        continue
    else:
        print((li.index(x) +1) / x)
```

we just want to skip the number to avoid dividing by 0

LISTS AND LOOP PRACTICE

- Practice using continue and break together with loops and lists.
  - Make a list of strings. We want to find the first string that is longer than 5 letters. If the length of the string is longer than 5, stop the loop.
  - Make a list of integers. Divide them by 8. If the remainder is 0, go to the next number. If it is not 0, add the number to a new list.