J233
Coding for Journalists

LECTURER
Soo Oh

PROMPTS

Sign into Slack and Google

start Zoom recording

Agenda

Announcements + check-in

Homework review + how much time

Pop quiz

Control flow

BREAK

Homework

Announcements and checking in

How was Saturday homework?

Final project ideas

Codecademy

Homework Review

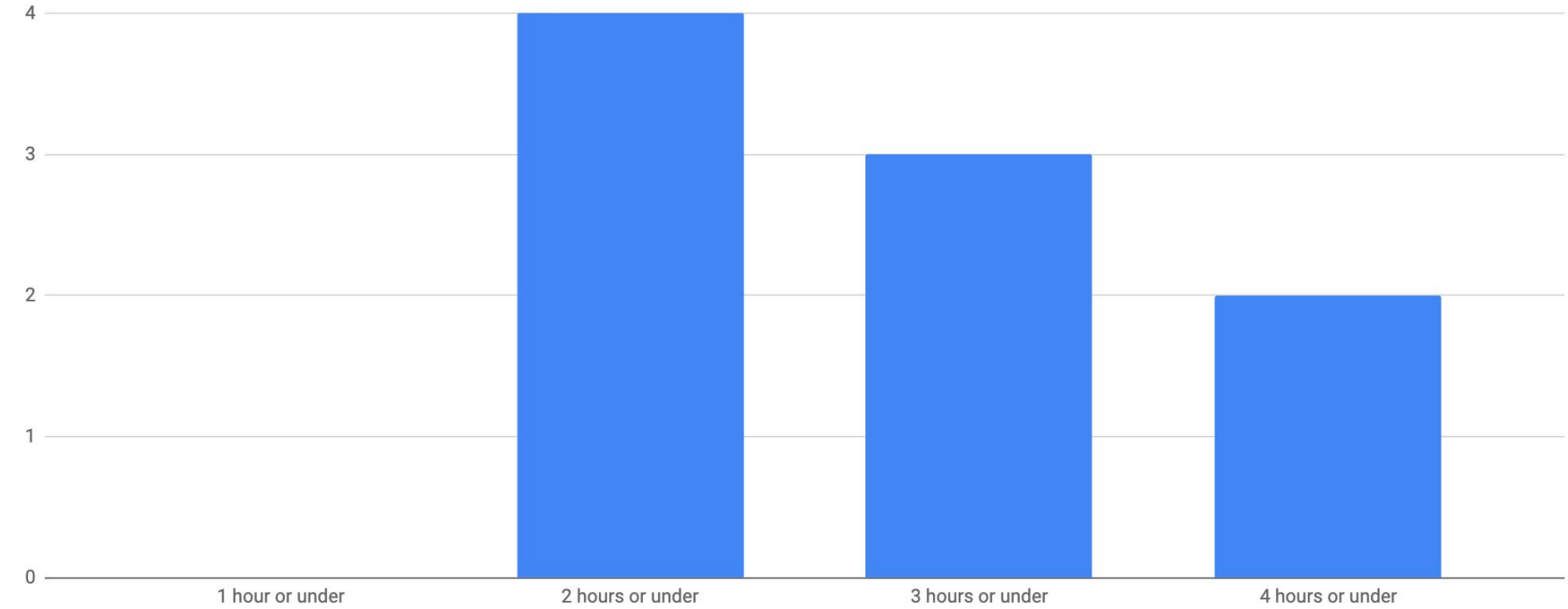
Documenting

Markdown

You got credit for using Markdown blocks. For future notebooks, include:

- Assignment name in title
- Your name somewhere at the top
- *Some* formatting for the questions, preferably using header styles

Week of 0918: students grouped by time spent outside of lecture and office hours



What questions do you have?

Control flow

`Control flow`

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

Types of **control flow**:

- **sequential** (what we've been doing so far, we execute one line after another in order)
- **selection** (`if`... `elif`... `else`)
- **repetition** (`for` and `while` loops)

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Let's say who know someone named Alex
alex_age = 13
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Let's say who know someone named Alex
alex_age = 13

if alex_age < 16:
    alex_can_drive_legally = False
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Let's say who know someone named Alex
alex_age = 13

if alex_age < 16:
    alex_can_drive_legally = False
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Let's say who know someone named Alex
alex_age = 13

if alex_age < 16:
    alex_can_drive_legally = False
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Let's say who know someone named Alex
alex_age = 13

if alex_age < 16:
    alex_can_drive_legally = False
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Let's say who know someone named Alex
alex_age = 13

if alex_age < 16:
    alex_can_drive_legally = False
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Let's say who know someone named Alex
alex_age = 13

if alex_age < 16:
    alex_can_drive_legally = False
else:
    alex_can_drive_legally = True
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Let's say who know someone named Alex
alex_age = 13

if alex_age < 16:
    alex_can_drive_legally = False
else:
    alex_can_drive_legally = True
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Let's say who know someone named Alex
alex_age = 13

if alex_age < 16:
    alex_can_drive_legally = False
else:
    alex_can_drive_legally = True

alex_can_drive_legally
Out[]:
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Let's say who know someone named Alex
alex_age = 13

if alex_age < 16:
    alex_can_drive_legally = False
else:
    alex_can_drive_legally = True

alex_can_drive_legally
Out[]: False
```

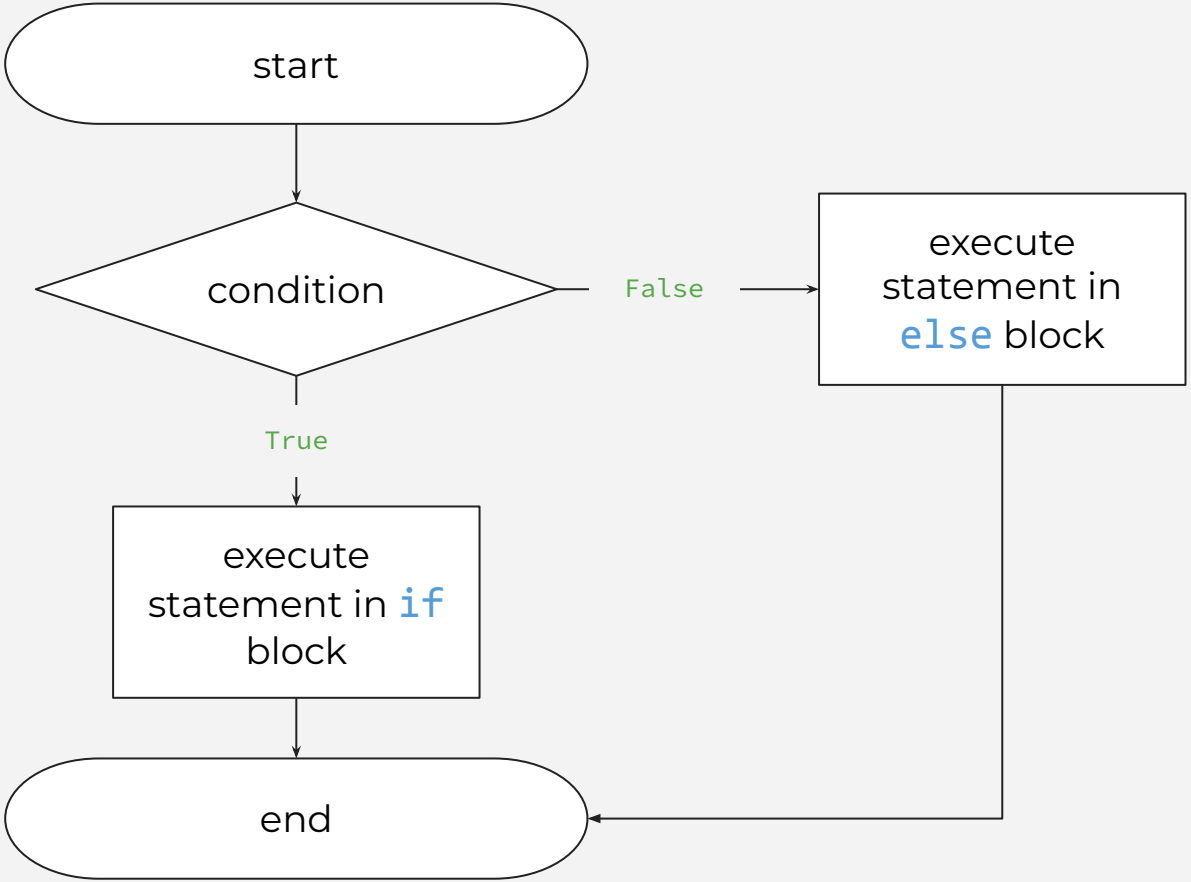
Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops



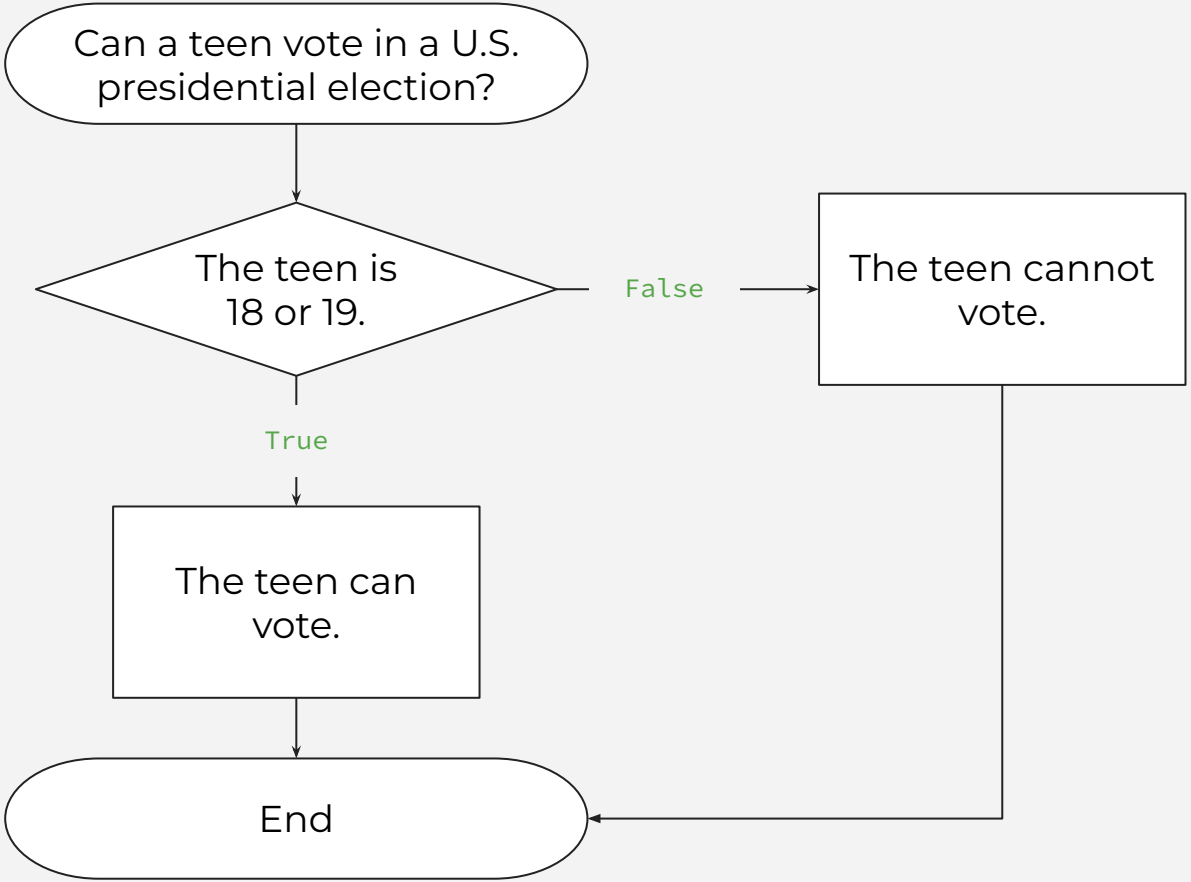
Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops



Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

> 15 human years equals the first year of a medium-sized dog's life.
>
> Year two for a dog equals about nine years for a human.
>
> And after that, each human year would be approximately five years for a dog.
>
> — American Kennel Club

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Write a function that takes human age
# and turns it into dog age.

dog_age(-1)
Out[]: 'Not born yet!'
dog_age(0)
Out[]: 0
dog_age(0.5)
Out[]: 'Enter an integer'
dog_age('puppy')
Out[]: 'Enter an integer'
dog_age(1)
Out[]: 15
dog_age(2)
Out[]: 24
dog_age(3)
Out[]: 29
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
def dog_age(human_age):
    pass
    # hints
    # you'll use type()
    # and if/elif/else statements

# What is `pass`? It's a null statement.
# Use pass instead of a comment when your
# code could break otherwise. For example,
# you might want to define a function but wait
# to write out what the function does until you
# set up code elsewhere.
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
def dog_age(human_age):
    if type(human_age) != int:
        return 'Enter an integer'
    elif human_age < 0:
        return 'Not born yet!'
    elif human_age == 0:
        return 0
    elif human_age == 1:
        return 15
    elif human_age == 2:
        return 24
    else:
        return 24 + (5 * (human_age - 2))
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
def dog_age(human_age):
    if type(human_age) != int:
        return 'Enter an integer'
    elif human_age < 0:
        return 'Not born yet!'
    elif human_age == 0:
        return 0
    elif human_age == 1:
        return 15
    elif human_age == 2:
        return 24
    else:
        return 24 + (5 * (human_age - 2))
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']

for pet in pets:
    print(pet)

Out[]:
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']

for pet in pets:
    print(pet)

Out[]: dog
cat
hamster
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']

for pet in pets:
    print(pet)

Out[]: dog
cat
hamster
```

What's the index of each item in the `pets` list?

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']
         pets[0]  pets[1]  pets[2]

for pet in pets:
    print(pet)

Out[]: dog
cat
hamster
```

What's the index of each item in the `pets` list?

Control flow

Conditionals (`if`...)

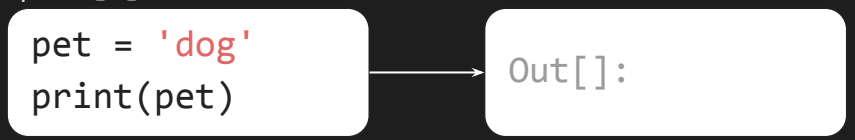
`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']
          pets[0]

for pet in pets:
    print(pet)
```



Control flow

Conditionals (`if`...)

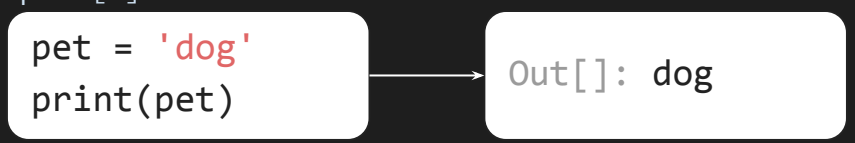
`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']
          pets[0]

for pet in pets:
    print(pet)
```



Control flow

Conditionals (`if`...)

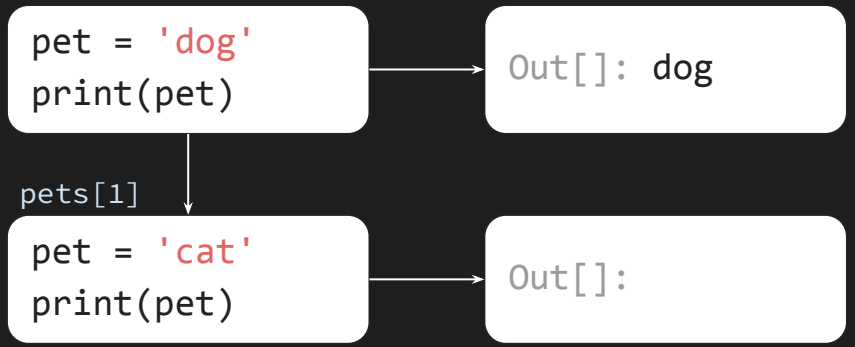
`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']
                pets[1]

for pet in pets:
    print(pet)
```



Control flow

Conditionals (`if`...)

`for` loops

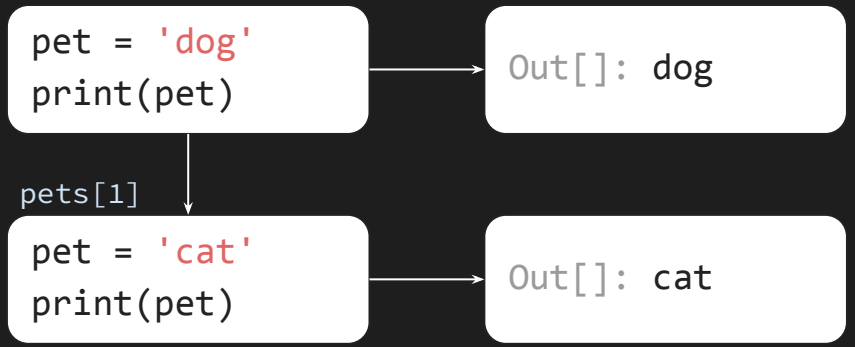
`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']
```

pets[1]

```
for pet in pets:
    print(pet)
```



Control flow

Conditionals (`if`...)

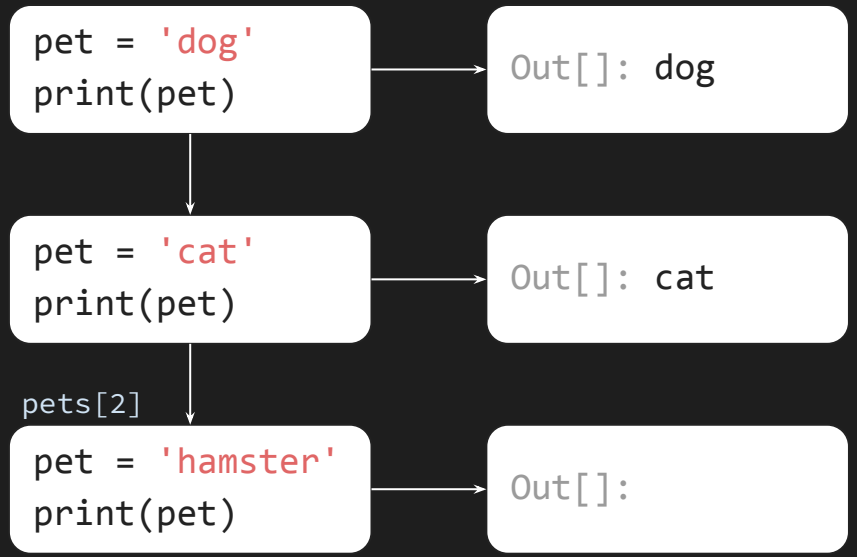
`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']
                      pets[2]

for pet in pets:
    print(pet)
```



Control flow

Conditionals (`if`...)

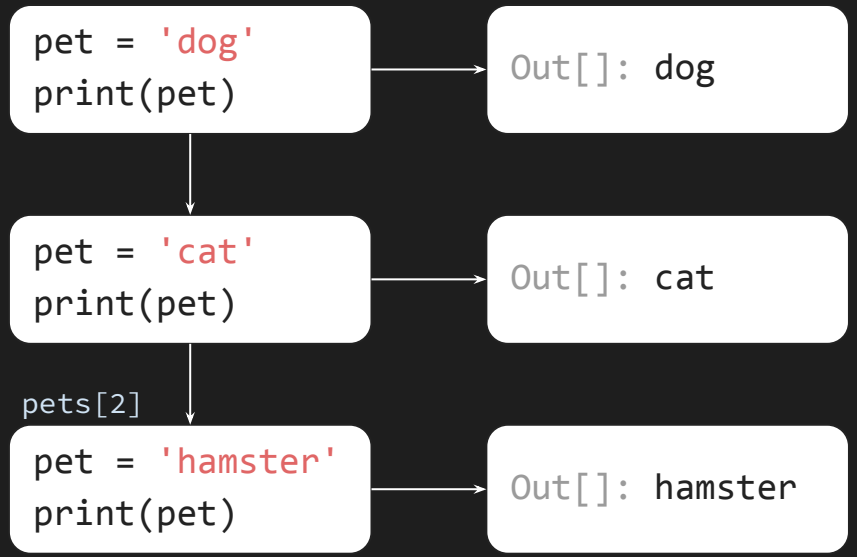
`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']
                      pets[2]

for pet in pets:
    print(pet)
```



Control flow

Conditionals (`if`...)

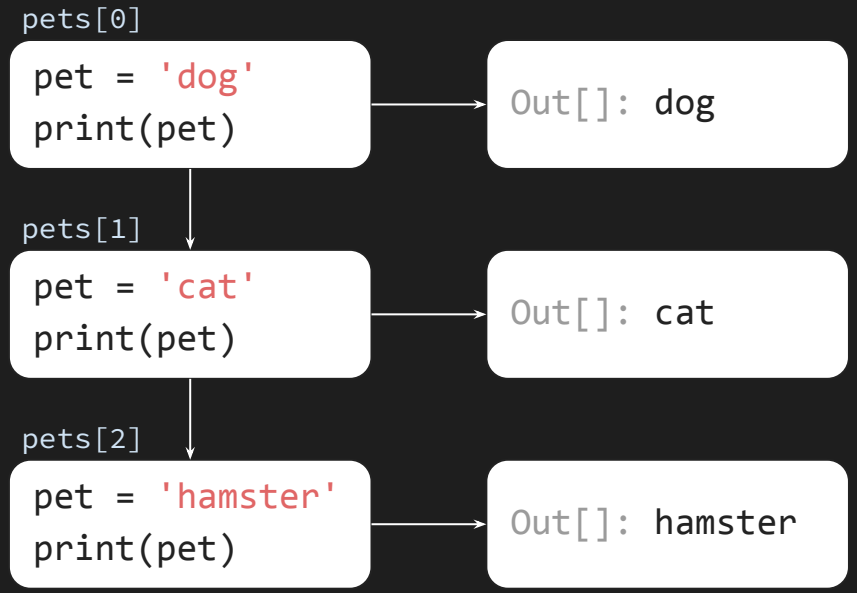
`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']
         pets[0]  pets[1]  pets[2]

for pet in pets:
    print(pet)
```



Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']

for pet in pets:
    print(pet)

Out[]: dog
cat
hamster
```

Control flow

Conditionals (`if`...)

`for` loops

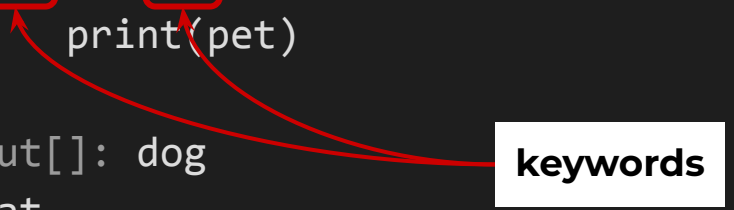
`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']

for pet in pets:
    print(pet)

Out[]: dog
cat
hamster
```



Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']

for pet in pets:
    print(pet)

Out[]: dog
cat
hamster
```

iterator (doesn't have to be a list)

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
pets = ['dog', 'cat', 'hamster']

for pet in pets:
    print(pet)

Out[]: dog
cat
hamster
```

variable name of your choice

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]

for n in nums:
    pass
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n

print(sum)
Out[]: 15
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n
```

`n = 1` `n = 2` `n = 3` `n = 4` `n = 5`

Control flow

Conditionals (`if`...)

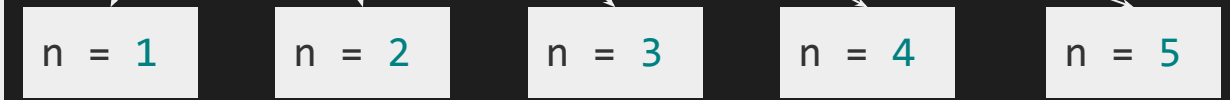
`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n
```



Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n
```

`n = 1` `n = 2` `n = 3` `n = 4` `n = 5`

```
sum = 0
n = 1
sum = sum + n
```

statement inside `for` loop

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n
```

`n = 1` `n = 2` `n = 3` `n = 4` `n = 5`

```
sum = 0
n = 1
sum = sum + n
print(sum)
Out[]:
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n
```

`n = 1` `n = 2` `n = 3` `n = 4` `n = 5`

```
sum = 0
n = 1
sum = sum + n
print(sum)
Out[]: 1
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n
```

`n = 1` `n = 2` `n = 3` `n = 4` `n = 5`

```
sum =
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n
```

`n = 1` `n = 2` `n = 3` `n = 4` `n = 5`

```
sum = 1
```

`sum` is outside the `for` loop!

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n
```

`n = 1` `n = 2` `n = 3` `n = 4` `n = 5`

```
sum = 1
n = 2
sum = sum + n
```

`sum` is outside the `for` loop!

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n
```

`n = 1` `n = 2` `n = 3` `n = 4` `n = 5`

```
sum = 1
n = 2
sum = sum + n
print(sum)
Out[]:
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n
```

`n = 1` `n = 2` `n = 3` `n = 4` `n = 5`

```
sum = 1
n = 2
sum = sum + n
print(sum)
Out[]: 3
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n
```

`n = 1` `n = 2` `n = 3` `n = 4` `n = 5`

```
sum = 3
n = 3
sum = sum + n
print(sum)
Out[]: 6
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n
```

`n = 1` `n = 2` `n = 3` `n = 4` `n = 5`

```
sum = 6
n = 4
sum = sum + n
print(sum)
Out[]: 10
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n
```

`n = 1` `n = 2` `n = 3` `n = 4` `n = 5`

```
sum = 10
n = 5
sum = sum + n
print(sum)
Out[]: 15
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# How can we add up the sum of the items in nums?

nums = [1, 2, 3, 4, 5]
sum = 0
for n in nums:
    sum = sum + n

print(sum)
Out[]: 15
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# loop through a dict
store = {'name': 'Berkeley General Store',
'apples': 52, 'bananas': 9, 'kiwis': 27}

for key, value in store.items():
    pass


Out[]:
I called Berkeley General Store.
The store has 52 apples in stock.
The store has 9 bananas in stock.
The store has 27 kiwis in stock.
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# loop through a dict
store = {'name': 'Berkeley General Store',
'apples': 52, 'bananas': 9, 'kiwis': 27}

for key, value in store.items():
    if key == 'name':
        print(f'I called { store[key] }.')
    else:
        print(f'They have { store[key] } {key}.')

Out[]:
I called Berkeley General Store.
The store has 52 apples in stock.
The store has 9 bananas in stock.
The store has 27 kiwis in stock.
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

How to use `range()`

```
# increment by 1, starting from 0 to 5
# but not including 5
for n in range(5):
    print(n)

# decrement by 1, starting at 5 until 0
# but not including 0
for n in range(5, 0, -1):
    print(n)
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Create a function that calculates the factorial
# of a number. The factorial function says to
# multiply all whole numbers from a given number
# down to 1.
# For example:
# 4! = 4 × 3 × 2 × 1 = 24

factorial(4)
Out[]: 24
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
def factorial(n):
    result = 1
    for x in range(n, 0, -1):
        result = result * x
    return result
```

Break

Meet back in 15 minutes

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# While the condition is met
# continue to loop
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# While the condition is met
# continue to loop
def question():
    answer = 'n'
    while answer == 'n':
        answer = input('Do you like coding now?
        Type "y" for yes or "n" for no.')

question()
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# While the condition is met
# continue to loop
def question():
    answer = 'n'
    while answer == 'n':
        answer = input('Do you like coding now?
        Type "y" for yes or "n" for no.')

question()
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Write a while loop that prints
# out numbers from 1 to 10
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# Write a while loop that prints
# out numbers from 1 to 10
n = 0
while n < 10:
    n = n + 1
    print(n)
```

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

Avoid infinite loops!

- `for` loops end (eventually)
- `while` loops might not end

Control flow

Conditionals (`if`...)

`for` loops

`while` loops

infinite loops

```
# What's wrong with this loop?

n = 1

while (n <= 1):
    print(n)
```

Questions?
Exercises?

Homework

https://journ233.github.io