

# Dictionaries, JSON, and Pip



**Sarah Holderness**

Author

@dr\_holderness



# Maintaining Two Lists

```
acronyms = ['LOL', 'IDK', 'TBH']  
translations = ['laugh out loud', 'I don't know', 'to be honest']
```

```
del acronyms[0]  
del translations[0]
```

◀...

*If we add or delete from one list ...  
We have to do the same thing in the other list.*

```
print(acronyms)  
print(translations)
```

```
> ['IDK', 'TBH']  
  ['I don't know', 'to be honest']
```

# A Dictionary Maps Keys to Values

```
acronyms = {'LOL': 'laugh out loud',  
            'IDK': "I don't know",  
            'TBH': 'to be honest'}
```



Key	Value
'LOL'	'laugh out loud'
'IDK'	"I don't know"
'TBH'	to be honest'

These would be the keys and values stored in the dictionary.

Each item is known as a "key-value pair"

# A Dictionary Maps Keys to Values

```
acronyms = {'LOL': 'laugh out loud',  
            'IDK': "I don't know",  
            'TBH': 'to be honest'}
```

```
print(acronyms['LOL'])
```

◀... To look up a *value* in a dictionary,  
we send in a *key*.  
(Instead of an index in a list.)

```
> laugh out loud
```

# Dictionaries Can Hold Anything

## Dictionary of strings to strings

```
acronyms = {'LOL': 'laugh out loud', 'IDK': "I don't know"}
```

## Dictionary of strings to numbers

```
menu = {'Soup': 5, 'Salad': 6}
```

*A menu item's name is the key,  
and its price is the value.*

## Dictionary of anything

```
my_dict = {10: 'hello', 2: 6.5}
```

# Creating a Dictionary and Adding Values

```
acronyms = {}
```

←... Create an empty dictionary

```
acronyms['LOL'] = 'laugh out loud'
```

```
acronyms['IDK'] = "I don't know"
```

```
acronyms['TBH'] = 'to be honest'
```

←... Adding new dictionary items

```
print(acronyms)
```

```
> {'IDK': "I don't know", 'LOL': 'laugh out loud',  
   'TBH': 'to be honest'}
```

←... Notice our 3 key-value pairs are there, but order is random in a dictionary.

# Updating Values in Our Dictionary

```
acronyms = {'LOL': 'laugh out loud',  
            'IDK': "I don't know",  
            'TBH': 'to be honest'}
```

... We can also create a dictionary with initial values

```
acronyms['TBH'] = 'honestly'
```

... A value is update the same way a value is added.

```
print(acronyms['TBH'])
```

... Looking up the same value

```
> honestly
```

... The value is printed

# Removing Dictionary Items

```
acronyms = {'LOL': 'laugh out loud',  
            'IDK': "I don't know",  
            'TBH': 'to be honest'}
```

```
del acronym['LOL']
```

◀... *To delete a value in a dictionary, we send in a key just like when we look up a value.*

```
print(acronyms)
```

```
> {'IDK': "I don't know", 'TBH': 'to be honest'}
```



# Getting an Item That's NOT in the Dictionary

```
acronyms = {'LOL': 'laugh out loud',  
            'IDK': "I don't know",  
            'TBH': 'to be honest'}
```

```
definition = acronyms['BTW']
```



*Trying to access a key that doesn't exist will cause an error*

```
> KeyError: 'BTW'
```

# Getting an Item That's NOT in the Dictionary

```
acronyms = {'LOL': 'laugh out loud',  
            'IDK': "I don't know",  
            'TBH': 'to be honest'}
```

```
definition = acronyms.get('BTW') ◀...
```

*Using get() won't crash your program with an error.*

```
print(acronyms) ◀...
```

*Instead, get() will return None if the key doesn't exist*

```
> None ◀...
```

*None is a type that represents the absence of a value.*

# None Type

None means the absence of a value, and values to False in a conditional

```
acronyms = {'LOL': 'laugh out loud', 'IDK': "I don't know"}  
definition = acronyms.get('BTW') ◀...
```

*Definition equals None because the key doesn't exist*

```
if definition: ◀...  
    print(definition)
```

*False because definition is None*

```
else:  
    print("Key doesn't exist") ◀...
```

*So this is run*

```
> Key doesn't exist
```

# Using a Dictionary to Translate a Sentence

```
acronyms = {'LOL': 'laugh out loud',  
            'IDK': "I don't know",  
            'TBH': 'to be honest'}
```

```
sentence = 'IDK' + ' what happened ' + 'TBH'
```

```
translation = acronym.get('IDK') + ' what happened ' + acronym.get('TBH')
```



*Look up the value of each acronym  
in the acronym dictionary*

# Using a Dictionary to Translate a Sentence

```
acronyms = {'LOL': 'laugh out loud',  
            'IDK': "I don't know",  
            'TBH': 'to be honest'}  
  
sentence = 'IDK' + ' what happened ' + 'TBH'  
translation = acronym.get('IDK') + ' what happened ' + acronym.get('TBH')  
  
print('sentence:', sentence)  
print('translation:', translation)
```

```
> sentence: IDK what happened TBH  
   translation: I don't know what happened to be honest
```

Up Next:

# Demo: Create a Movie Schedule Dictionary

---



# Combining Lists and Dictionaries



**Sarah Holderness**

Author

@dr\_holderness



# Breakfast, Lunch and Dinner Lists

Let's say we have three separate menu lists: Breakfast, Lunch, Dinner...

```
breakfast = ['Egg Sandwich', 'Bagel', 'Coffee']
```



```
lunch = ['BLT', 'PB&J', 'Turkey Sandwich']
```



```
dinner = ['Soup', 'Salad', 'Spaghetti', 'Taco']
```



*How would we combine these  
into one list?*



# A Lists of Lists

You can have a container of containers – menus is a list of lists

```
menus = [ ['Egg Sandwich', 'Bagel', 'Coffee'],  
          ['BLT', 'PB&J', 'Turkey Sandwich'],  
          ['Soup', 'Salad', 'Spaghetti', 'Taco'] ]
```

```
print('Breakfast Menu:\t', menus[0])  ◀... 1st list  
print('Lunch Menu:\t',      menus[1])  ◀... 2nd list  
print('Dinner Menu:\t',     menus[2])  ◀... 3rd list
```

```
> Breakfast Menu: ['Egg Sandwich', 'Bagel', 'Coffee']  
  Lunch Menu:     ['BLT', 'PB&J', 'Turkey Sandwich']  
  Dinner Menu:    ['Soup', 'Salad', 'Spaghetti', 'Taco']
```

# Getting an Item from a 2-dimensional List

You can use 2 indexes to get an individual item from an inner list

```
menus = [ ['Egg Sandwich', 'Bagel', 'Coffee'],  
          ['BLT', 'PB&J', 'Turkey Sandwich'],  
          ['Soup', 'Salad', 'Spaghetti', 'Taco'] ]
```

*How would you access this item?*

```
print(menus[0])
```

*1st list*

```
> ['Egg Sandwich', 'Bagel', 'Coffee']
```

*A good start! Now we just need  
to get the second item in this list*

# Getting an Item from a 2-dimensional List

You can use 2 indexes to get an individual item from an inner list

```
menus = [ ['Egg Sandwich', 'Bagel', 'Coffee'],  
          ['BLT', 'PB&J', 'Turkey Sandwich'],  
          ['Soup', 'Salad', 'Spaghetti', 'Taco'] ]
```



*How would you access this item?*

```
print(menus[0][1])
```



*The list at index 0 in the outer list,  
the item at index 1 in the inner list*

```
> Bagel
```

# A Dictionary of Lists

We could also use a dictionary for our menus with keys for Breakfast, Lunch, and Dinner

```
menus = { 'Breakfast' : ['Egg Sandwich', 'Bagel', 'Coffee'],  
          'Lunch'      : ['BLT', 'PB&J', 'Turkey Sandwich'],  
          'Dinner'     : ['Soup', 'Salad', 'Spaghetti', 'Taco'] }
```

```
print('Breakfast Menu:\t', menus['Breakfast'])
```

```
print('Lunch Menu:\t', menus['Lunch'])
```

```
print('Dinner Menu:\t', menus['Dinner'])
```

*Using the keys to  
access each list*

```
> Breakfast Menu: ['Egg Sandwich', 'Bagel', 'Coffee']  
Lunch Menu:      ['BLT', 'PB&J', 'Turkey Sandwich']  
Dinner Menu:     ['Soup', 'Salad', 'Spaghetti', 'Taco']
```

# A Dictionary of Lists

We could also use a dictionary for our menus with keys for Breakfast, Lunch, and Dinner

```
menus = { 'Breakfast' : ['Egg Sandwich', 'Bagel', 'Coffee'],  
          'Lunch'      : ['BLT', 'PB&J', 'Turkey Sandwich'],  
          'Dinner'     : ['Soup', 'Salad', 'Spaghetti', 'Taco'] }
```

```
print('Breakfast Menu:\t', menus['Breakfast'])
```

```
print('Lunch Menu:\t', menus['Lunch'])
```

```
print('Dinner Menu:\t', menus['Dinner'])
```

*What if we had a lot more menus, is there a better way to print each one?*

```
> Breakfast Menu:  ['Egg Sandwich', 'Bagel', 'Coffee']  
Lunch Menu:       ['BLT', 'PB&J', 'Turkey Sandwich']  
Dinner Menu:      ['Soup', 'Salad', 'Spaghetti', 'Taco']
```

# Printing the Dictionary Menu Items

```
menus = { 'Breakfast' : ['Egg Sandwich', 'Bagel', 'Coffee'],  
          'Lunch'      : ['BLT', 'PB&J', 'Turkey Sandwich'],  
          'Dinner'     : ['Soup', 'Salad', 'Spaghetti', 'Taco'] }
```

```
for item in menus:  
    print(item)
```

*This defaults to just returning  
the keys in a dictionary...  
Which is not what we want.*

```
> Breakfast  
Lunch  
Dinner
```

# Using a Dictionary's Key and Value in a for Loop

```
menus = { 'Breakfast' : ['Egg Sandwich', 'Bagel', 'Coffee'],  
          'Lunch'      : ['BLT', 'PB&J', 'Turkey Sandwich'],  
          'Dinner'     : ['Soup', 'Salad', 'Spaghetti', 'Taco'] }
```

```
for name, menu in menus.items():  
    print(name, ':', menu)
```

*Now the loop has  
access to both the key  
and the value here.*

```
> Breakfast: ['Egg Sandwich', 'Bagel', 'Coffee']  
Lunch:      ['BLT', 'PB&J', 'Turkey Sandwich']  
Dinner:     ['Soup', 'Salad', 'Spaghetti', 'Taco']
```

# Using Dictionaries to Represent Objects



*Let's say we have a person and we want to represent their attributes, such as their name, age, and city they're from.*

```
person = {'name': 'Sarah Smith',  
          'city': 'Orlando',  
          'age': '100'}
```

*We could use a dictionary where the attributes are saved as key, value pairs.*

```
print(person.get('name'), 'is', person.get('age'), 'years old.')
```

```
> Sarah Smith is 100 years old.
```



Up Next:

# Demo:

# Parse a Nested Contacts Dictionary

---



# Reading JSON and Installing Packages with Pip



**Sarah Holderness**

Author

@dr\_holderness



# We Want a Program that Lists the Current People in Space

```
>>> python3 space.py
```

The people currently  
in space are:

Sergey Prokopyev

Dmitry Petelin

Frank Rubio

Nicole Mann

Josh Cassada

Koichi Wakata

Anna Kikina

Fei Junlong

Deng Qingming

Zhang Lu

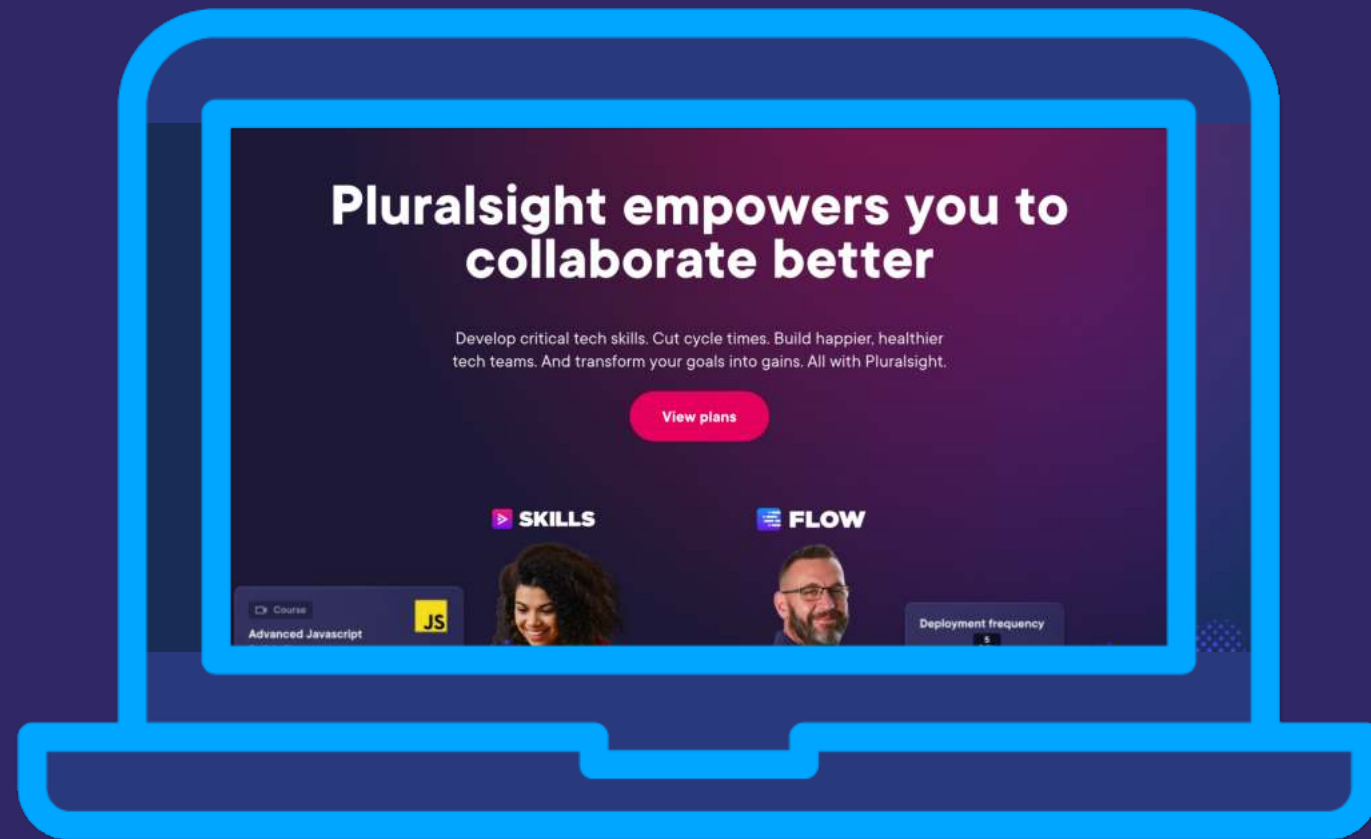
HTTP Request to  
[api.open-notify.org/astros.json](https://api.open-notify.org/astros.json)



HTTP Response with  
the current people  
in space



# HTTP Request



Your Computer

HTTP Request to  
[pluralsight.com](https://pluralsight.com)



HTTP Response

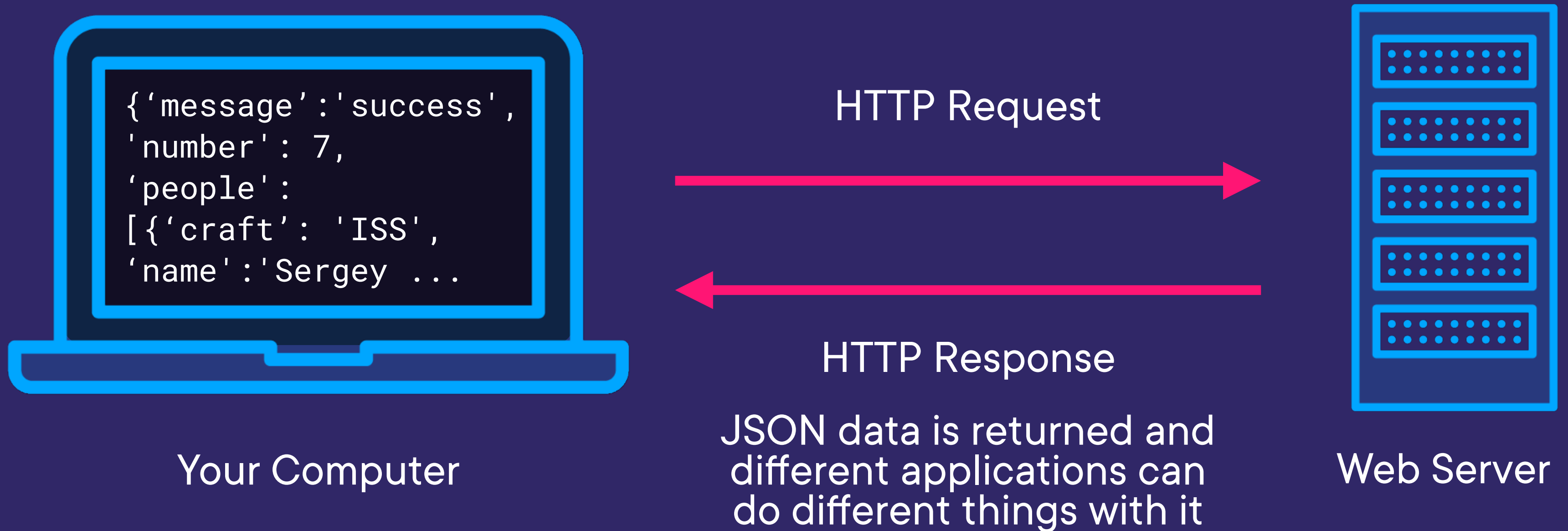
HTML can be returned  
and displayed as a web  
page in your browser



Web Server

# Some Websites Return Raw Data

Usually the raw data is returned under the API (Application Programming Interface) for the website such as [api.twitter.com](https://api.twitter.com)



# JSON Data

A common use of JSON is to exchange data to/from a web server

```
json = {  
    "number": 4,  
    "students":  
        [  
            {"name": "Sarah Holderness", "email": "sarah@example.com"},  
            {"name": "Harry Potter", "email": "harry@example.com"},  
            {"name": "Hermione Granger", "email": "hermione@example.com"},  
            {"name": "Ron Weasley", "email": "ron@example.com"}  
        ]  
    }  
}
```

◀.....

*JSON format can be a mix of lists and dictionaries like we've seen before*

# JSON – JavaScript Object Notation

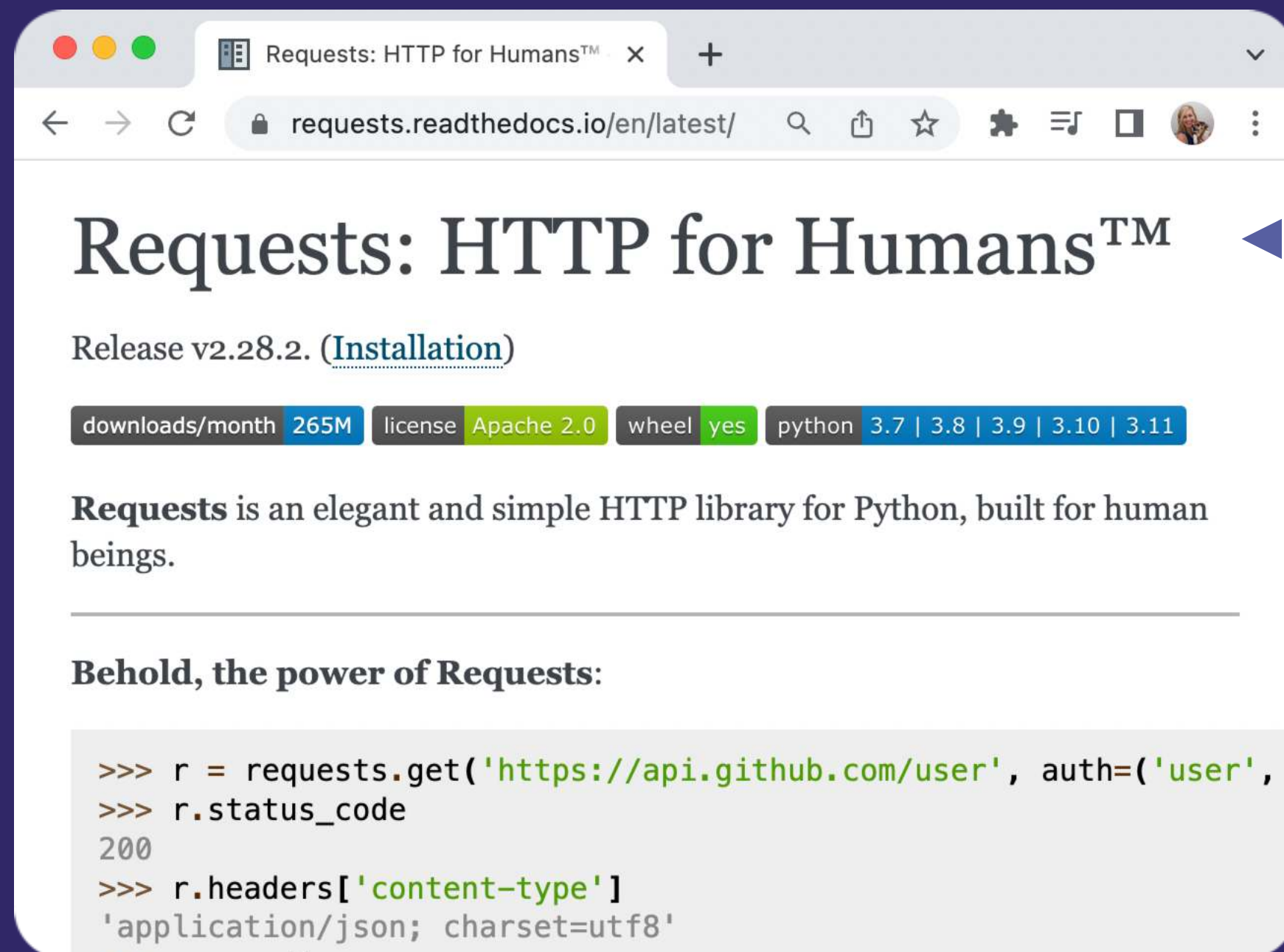
JSON started in JavaScript but now can be used in any programming language, including Python

```
json = {  
    "number":4,  
    "students":  
        [  
            {"name":"Sarah Holderness", "email":"sarah@example.com"},  
            {"name":"Harry Potter", "email":"harry@example.com"},  
            {"name":"Hermione Granger", "email":"hermione@example.com"},  
            {"name":"Ron Weasley", "email":"ron@example.com"}  
        ]  
}
```



# How Do We Do an HTTP Request in Python

If you do a web search for “Python http request” the 1st result should be the requests library



*The requests library allows us to do an http request.*

*However, requests is not installed with Python so we need to install it ourselves.*



# Ensure PIP is Installed

pip is used to install any package from the Python Package Index

*If Python 3 is installed the command `pip3` should work on a Mac or `pip` on Windows. (Otherwise add pip to the Path. )*

```
<<< pip3 --version
```

```
pip 22.3 from /Library/Frameworks/Python.framework/  
Versions/3.11/lib/python3.11/site-packages/pip  
(python 3.11)
```

# Installing the Requests Package

*You will need an internet connection since the packages are downloaded from the internet.*

```
<<< pip3 requests
```

```
Collecting requests
```

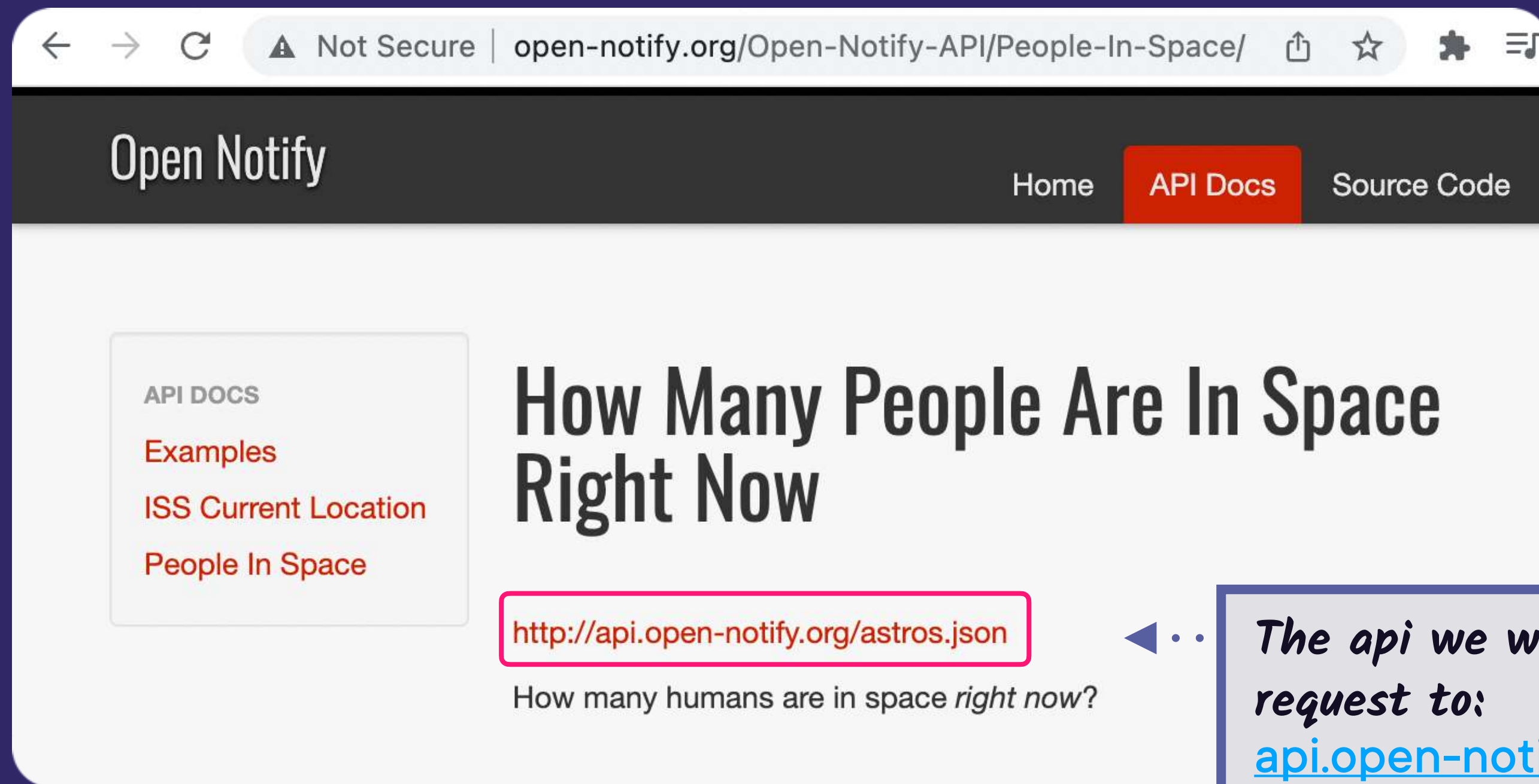
```
...
```

```
Successfully installed requests-2.28.2
```

*Now that the `requests` package is installed, we can use it in a program!*

# Where Do We Want to Request Data From?

[open-notify.org](https://open-notify.org) - This API returns the current number of people in space, and their names and spacecraft.



The screenshot shows a web browser window with the URL `open-notify.org/Open-Notify-API/People-In-Space/`. The page has a dark header with the 'Open Notify' logo and navigation links for 'Home', 'API Docs' (highlighted in red), and 'Source Code'. On the left, a sidebar lists 'API DOCS' with links to 'Examples', 'ISS Current Location', and 'People In Space'. The main content area features the heading 'How Many People Are In Space Right Now' and a red-bordered box containing the URL `http://api.open-notify.org/astros.json`. Below this box, the text 'How many humans are in space right now?' is visible.

*The api we want to send http the request to:*  
[api.open-notify.org/astros.json](http://api.open-notify.org/astros.json)

# How many humans are in space *right now* in JSON?

Making a request to [api.open-notify.org/astros.json](https://api.open-notify.org/astros.json) , will return the following JSON format.

```
{
  "message": "success",
  "number": NUMBER_OF_PEOPLE_IN_SPACE,
  "people": [
    { "name": NAME, "craft": SPACECRAFT_NAME },
    ...
  ]
}
```

# How Do We Do an HTTP Request in Python?

```
import requests
```



*First, we import the requests module*

# How Do We Do an HTTP Request in Python?

```
import requests
```

```
response = requests.get('http://api.open-notify.org/astros.json')
```



*Then we call `requests.get()` and pass in our http address.*

# How Do We Get JSON from the Response?

```
import requests
```

```
response = requests.get('http://api.open-notify.org/astros.json')  
json = response.json()
```



*Call `response.json()` to decode the JSON from the response.*

# Now We Can Print the JSON Data

```
import requests
```

```
response = requests.get('http://api.open-notify.org/astros.json')
```

```
json = response.json()
```

```
print(json)     ◀.....
```

*json stores the JSON as a dictionary*

```
> {  
    'message': 'success',  
    'number': 10  
    'people': [  
        {'name': 'Sergey Prokopyev', 'craft': 'ISS'},  
        {'name': 'Dmitry Peter', 'craft': 'ISS'},  
        ...  
    ]  
}
```



# Now We Can Print the JSON Data

```
import requests
```

```
response = requests.get('http://api.open-notify.org/astros.json')  
json = response.json()  
print(json)
```

```
> {  
  'message': 'success',  
  'number': 10  
  'people': [  
    {'name': 'Sergey Prokopyev', 'craft': 'ISS'},  
    {'name': 'Dmitry Peter', 'craft': 'ISS'},  
    ...  
  ]  
}
```

*How do we get this list of people?*

*We can use the people key:  
json['people']*

# Printing the People in Space

```
import requests
```

```
response = requests.get('http://api.open-notify.org/astros.json')  
json = response.json()
```

```
for person in json['people']:  
    print(person)
```

*We can use a loop to print each item in the list.*

```
> {'name': 'Sergey Prokopyev', 'craft': 'ISS'}  
   {'name': 'Dmitry Petelin', 'craft': 'ISS'}  
   ...
```

*Each item is a dictionary, but we just need the name part.*

# Printing the People in Space

```
import requests
```

```
response = requests.get('http://api.open-notify.org/astros.json')  
json = response.json()
```

```
for person in json['people']:  
    print(person['name'])
```

◀ ... We can use the key 'name' to get just the name.

```
> Sergey Prokopyev  
Dmitry Petelin  
Frank Rubio  
Nicole Mann  
Josh Cassada  
...
```

◀ ... Great! We're able to print the names of the people in space.

# Final Touch: Adding a Heading

```
import requests

response = requests.get('http://api.open-notify.org/astros.json')
json = response.json()

print('The people currently in space are:')
for person in json['people']:
    print(person['name'])
```

Print a heading  
for clarity.

> The people currently in space are:

Dmitry Petelin	Josh Cassada	Fei Junlong	Sergey Prokopyev
Frank Rubio	Koichi Wakata	Deng Qingming	
Nicole Mann	Anna Kikina	Zhang Lue	

Up Next:

**Demos:**

**Create a Python Virtual Environment &  
Use the Open Weather Map API**

---

