AI Bootcamp

Accessing APIs Securely

Module 6 Day 2

Class Objectives

By the end of class, you will be able to:

1. Explain the purpose of API keys.
2. Set/Export environment variables in Windows and Mac and retrieve them in Python.
3. Use API keys to fetch authenticated requests using the Requests Library.
4. Use **try** and **except** blocks to handle errors.

Activity:

API application—Requests and JSON Review

In this activity, you will review what you learned in Day 1.

Suggested Time:

10 Minutes

Time’s up!
Let’s review

Questions?

Instructor **Demonstration**

API Keys

API Keys

API keys are like keys to a house or car: they're used to get access to resources.

A key must be provided with every request for APIs that require keys.

API Keys

Companies use API keys as a means to secure data, as well as monitor traffic. Using keys in this manner allows companies to limit and block requests as needed.

API Keys

Obtaining an API key is like getting keys to a kingdom. Once you're in, you're empowered to build products and submit API requests as you please.

Instructor **Demonstration**

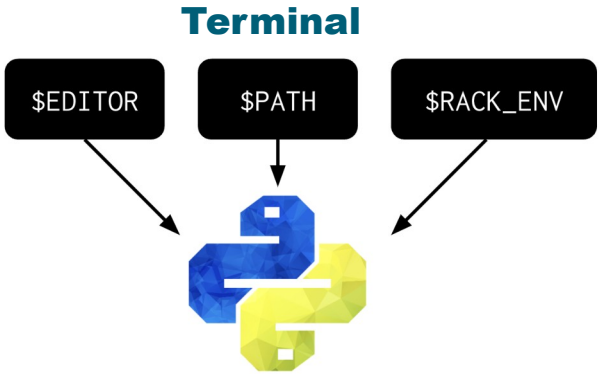
Creating Environment Variables

Environment Variables

Variables are accessed when stored as environment variables.

Environment Variables

Child process gets copies of parent's environment variables.



Local Variables

Child process doesn't get any copies of local variables.

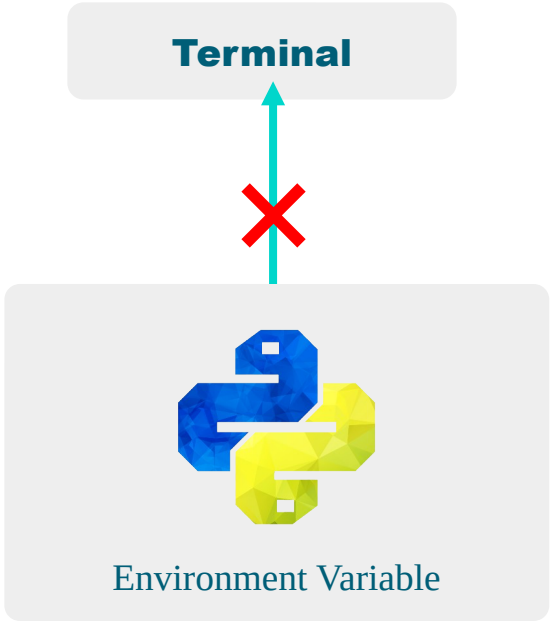
Environment Variables

A **.env** file can be used to put API keys into environment variables.

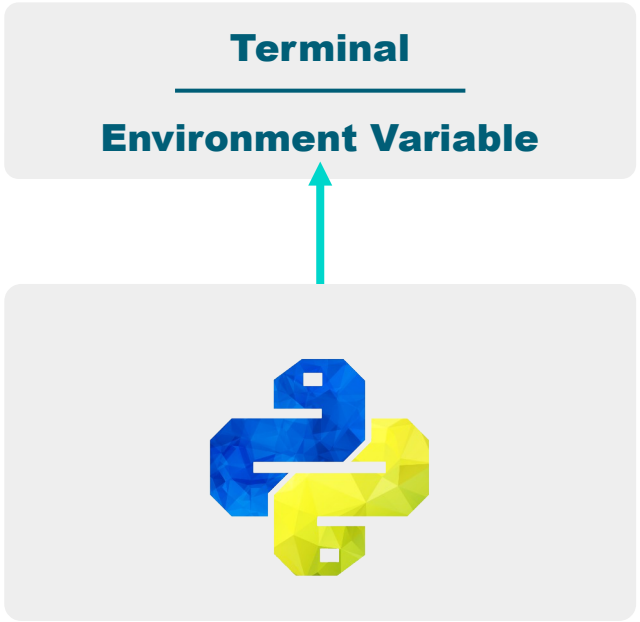
The **.env** file will contain the environment variable, and make it accessible by child processes.

Environment Variables

An environment variable created in Python cannot be accessed by a terminal.



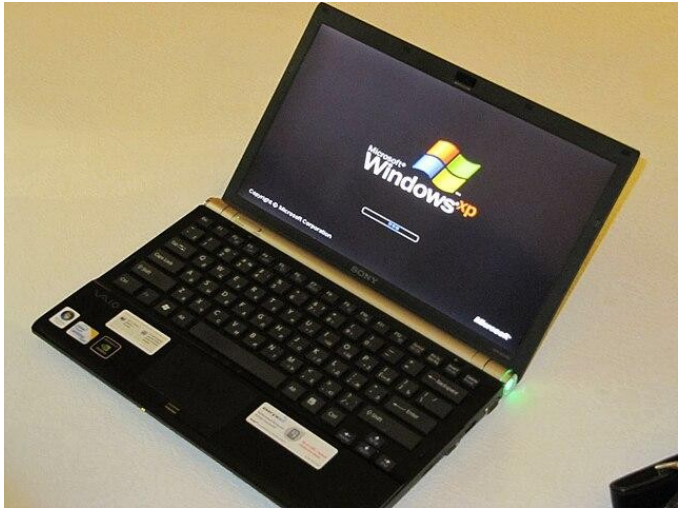
An environment variable created in a terminal can be accessed by Python.



Environment Variables

Because environment variables are at the operating system level, variables can be passed down from parent processes to child processes.

Apple



Windows

Instructor **Demonstration**

Calling Environment Variables

Calling Environment Variables

To make environment variables inheritable, they have to be exported and sourced.

```
api_key = os.getenv("API_KEY")
```

Activity:

Under Lock and Key

In this activity, you will create and use environment variables. You will also make an API request using an API key.

Suggested Time:

20 Minutes

Time’s up!
Let’s review

Questions?

Break

15 mins

Instructor **Demonstration**

New York Times API

Activity:

Retrieving Articles

In this activity, you will create access the New York Times API and make requests across multiple pages.

Suggested Time:

20 Minutes

Time’s up!
Let’s review

Questions?

Instructor **Demonstration**

OpenWeatherAPI

Activity:

Weather DataFrame

In this activity, you will retrieve data from an API with an API key and convert it to a DataFrame.

Suggested Time:

15 Minutes

Time’s up!
Let’s review

Questions?

Instructor **Demonstration**

Exception Handling

What would happen if an application tried to look up a key that doesn't exist within a given dictionary?

Errors

So far, our API requests have had the values we're looking for.

When a value is not found, Python returns an error,
as it does in our notebook when we look up **"Mary".**

```
students = {
    # Name  : Age
    "James": 27,
    "Sarah": 19,
    "Jocelyn": 28
}

print(students["Mary"])

print("This line will never print.")
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/var/folders/1n/50wf_pbs5bsb8__bwdkxwvq86mm3js/T/ipykernel_69817/95543560.py in
<module>
      6 }
      7
----> 8 print(students["Mary"])
      9
     10 print("This line will never print.")

KeyError: 'Mary'
```

Errors

The **try-except** code will let an application recover from errors like our Mary example.

- **"try"** and **except** are statements like **for** and **if.**
- Python will **"try"** to run the code.
- If the code throws an error or exception, the code in the **except** block is executed.

```
students = {
    # Name  : Age
    "James": 27,
    "Sarah": 19,
    "Jocelyn": 28
}

# Try to access key that doesn't exist
try:
    students["Mary"]
except KeyError:
    print("Oops, that key doesn't exist.")

# "Catching" the error lets the rest of our code execute
print("...But the program doesn't die early!")
```

```
Oops, that key doesn't exist.
...But the program doesn't die early!
```

Activity:

Making Exceptions

In this activity, you will use **try** and **except** to handle errors.

Suggested Time:

5 Minutes

Time’s up!
Let’s review

Questions?

Activity:

API Call Exceptions

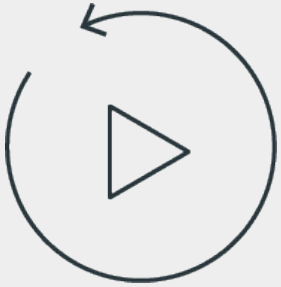
In this activity, you will use **try** and **except** in conjunction with API requests when query not found.

Suggested Time:

15 Minutes

Time’s up!
Let’s review

Questions?



Let's **recap**

Recap

After today's lesson, you are able to:

1. Explain the purpose of API keys.
2. Set/Export environment variables in Windows and Mac and retrieve them in Python.
3. Use API keys to fetch authenticated requests using the Requests Library.
4. Use `try` and `except` blocks to handle errors.

Next

The next lesson will culminate in a mini project that harnesses the power of APIs to build Python applications from scratch.

Questions?

The End