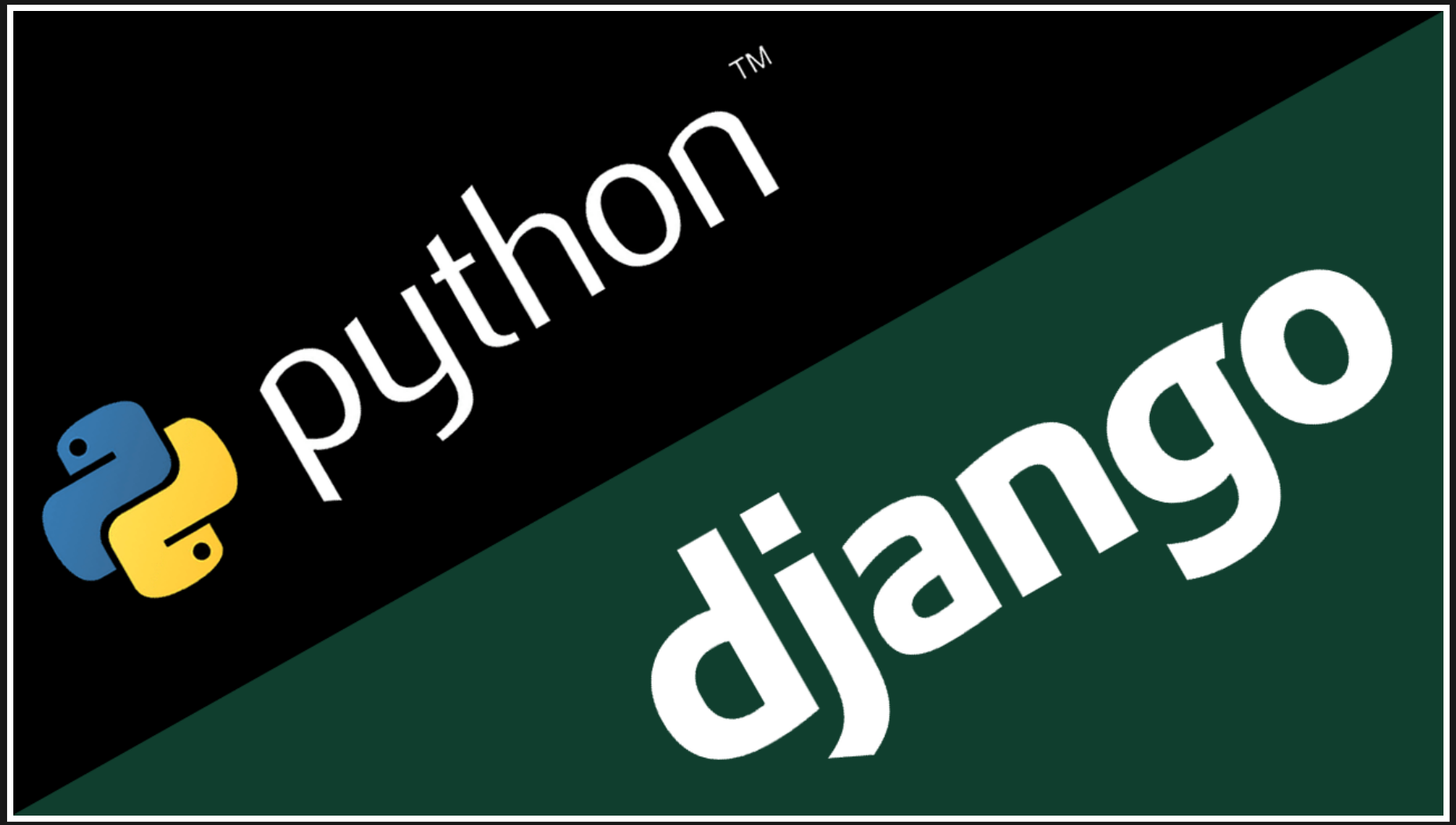


Patras Codecamp 2019

$ whoami

Dennis Rodis

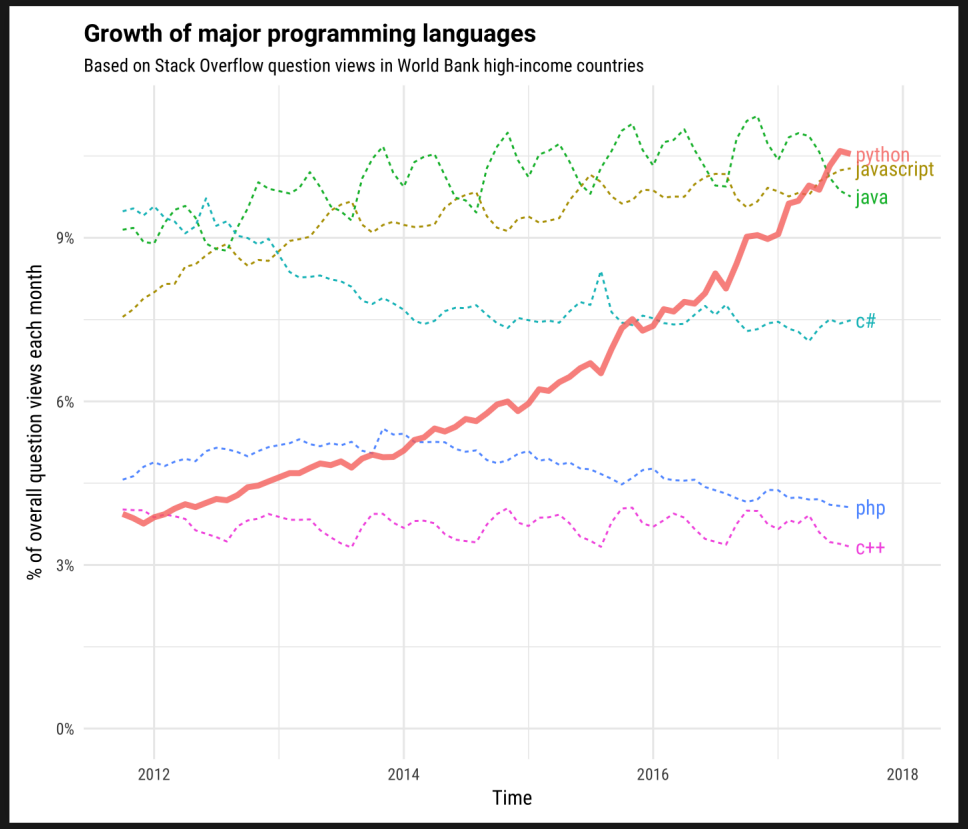
Software engineer / Web developer

Co Founder of Susurrus

Why Python

- Dynamic, Intepreted, Flexible
- High level
- Easy to use
- Easy to learn and read
- Well supported and popular
- Extensive Support Libraries
- Open source

Why Python - Popular



Hint: Python's popularity in data science and machine learning is probably the main driver of its fast growth.

The Incredible Growth of Python

Source: Stack Overflow Blog

Why Python - How is used

- Research
- Data Science, Machine Learning
- Information Security
- Internet of Things
- **Web development**

- Pre installed on Linux, Mac
- Python Releases for Windows

Python interpreter - Interactive mode

```
$ python
Python 3.5.2 (default, Sep 14 2017, 22:51:06)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more info
>>>
>>> print('hello world')
hello world
>>>
```

Python syntax

```
# integers
some_number = 10000

# booleans
true_boolean = True
false_boolean = False

# string
my_name = "Dennis"

# float
book_price = 15.80

type(some_number)
<class 'int'>
```

```
50 - 5*6         # 20
(50 - 5*6) / 4   # 5.0
8 / 5            # 1.6
5 ** 2           # 25

a = 5
a == 5           # True
a != 5           # False
a >= 5           # True

result = width * height
print(result)
```

If/Else

```
x = int(input('Pick a number: '))

if x < 0:
    print('Converting to positive...');
    x = -x
elif x == 0:
    print('Zero')
elif x == 1:
    print('One')
else:
    print('Greater than 1')
```

For loops

```
>>> for i in ['one', 'two', 'three']:
...     print(i)
...
one
two
three

>>> for num in range(10):
...     print(num, end=" ")
...
0 1 2 3 4 5 6 7 8 9
```

Functions

```
def find_min(a,b):
    if a < b:
        return a
    else:
        return b

print(find_min(3,4))

def greetings(greet="hello", name="there"):
    print(greet + ' ' + name)
```

Data Structures - Lists, Tuples

```
# Lists
a_list = []
whatever = ['abc', 4.56, [2,3], 'def', 6]
colors = ['red', 'green', 'blue', 'yellow']

print(colors[0])    # 'red'
print(colors[1])    # 'green'
len(colors)         # 4

colors.append('brown')
colors.remove('brown')

for color in colors:
    print(color.upper(), end=" ")    # RED GREEN BLUE YELLOW
```

Data Structures - Dictionaries

```
prices = { 'milk': 3.67, 'bread': 1.67, 'cheese': 4.67 }

print(prices)
print(prices['milk'])

prices['butter'] = 1.95
del prices['butter']

for k, v in prices.items():
    print(k, v)

# bread 1.67
# cheese 4.67
# milk 3.67
```

Classes

```
class Car:
    number_of_wheels = 4

    def __init__(self, fuel_type):
        self.type_of_tank = fuel_type

    def set_tank_type(self, type_of_tank):
        self.type_of_tank = type_of_tank

    def make_noise(self):
        print('VRUUUUUUUM')

tesla = Car('electric')
print(tesla.type_of_tank)
print(tesla.number_of_wheels)
```

Classes - Inheritance

```
class Vehicle:
    def __init__(self, number_of_wheels, fuel_type):
        self.number_of_wheels = number_of_wheels
        self.type_of_tank = fuel_type

class Car(Vehicle):
    def move(self):
        print('Moving...')

tesla = Car(4, 'electric')
print(tesla.number_of_wheels)
```

Import Python modules and packages

```
# Standard library imports
import datetime
import os

# Third party imports
from flask import Flask
from django.db import models

# Local application imports
from local_module import local_class
from local_package import local_function

# Relative imports
from .some_module import some_class
from . import some_class
```

Meet Django

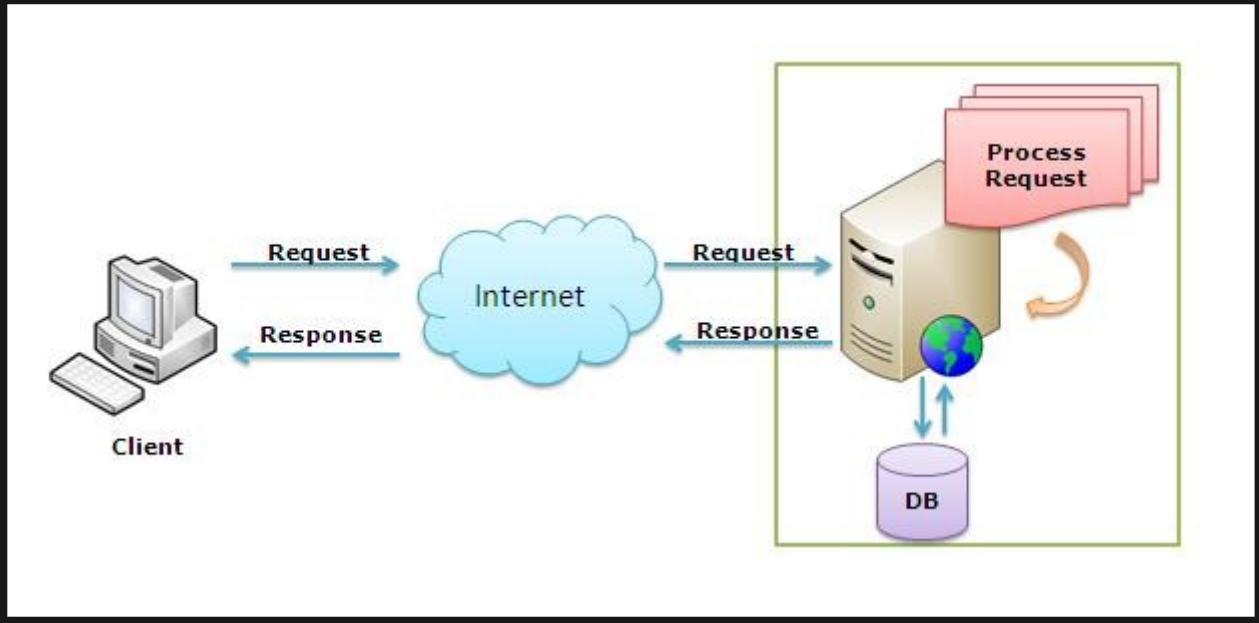
Django is a free and open source web application framework, written in Python. A web framework is a set of components that helps you to develop websites faster and easier.

Why Django

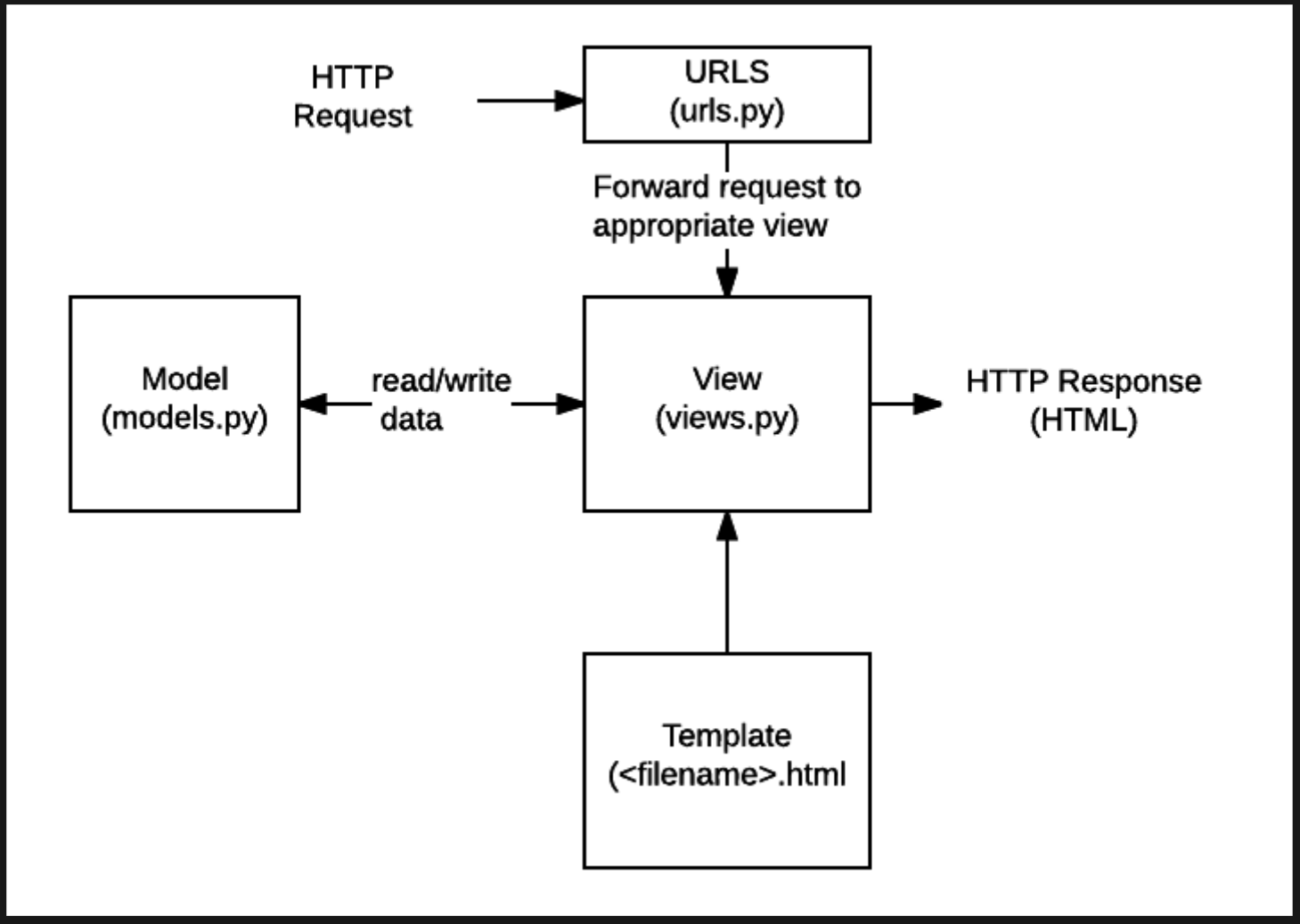
- Fully loaded (Batteries included)
- Powerful Admin panel
- Rich ecosystem
- Secure
- Get started quickly (rapid development)

Provided by Django

- Interacting with databases (ORM)
- Template system
- HTTP libraries
- Administration site
- User authentication and permission
- Forms handling
- Serialising data
- Caching


Process
Request
Request
Response
Internet
Request
Response
Client
DB

Architecture



Try Django

https://repl.it/languages/django

Notes

https://github.com/dencorg/quotes/blob/master/NOTES