Python para Devs JS

Jefferson Moura

Front-end Engineer @ Conta Utvikling

@jeffersonmourak

evolux

CALL CENTER ACD & ANALYTICS

Estão contratando!

$$$$$$$ Jabá $$$$$$$


Sim

É Sim

Eu Amo JS <3

JavaScript Developers


Então pq to falando
de Python?
WHY?

Por que Digamos que o JS não resolve tudo...

VS

Pelo menos não tão rápido!

Quanto o python!

JS


Então resolvi aprender Python

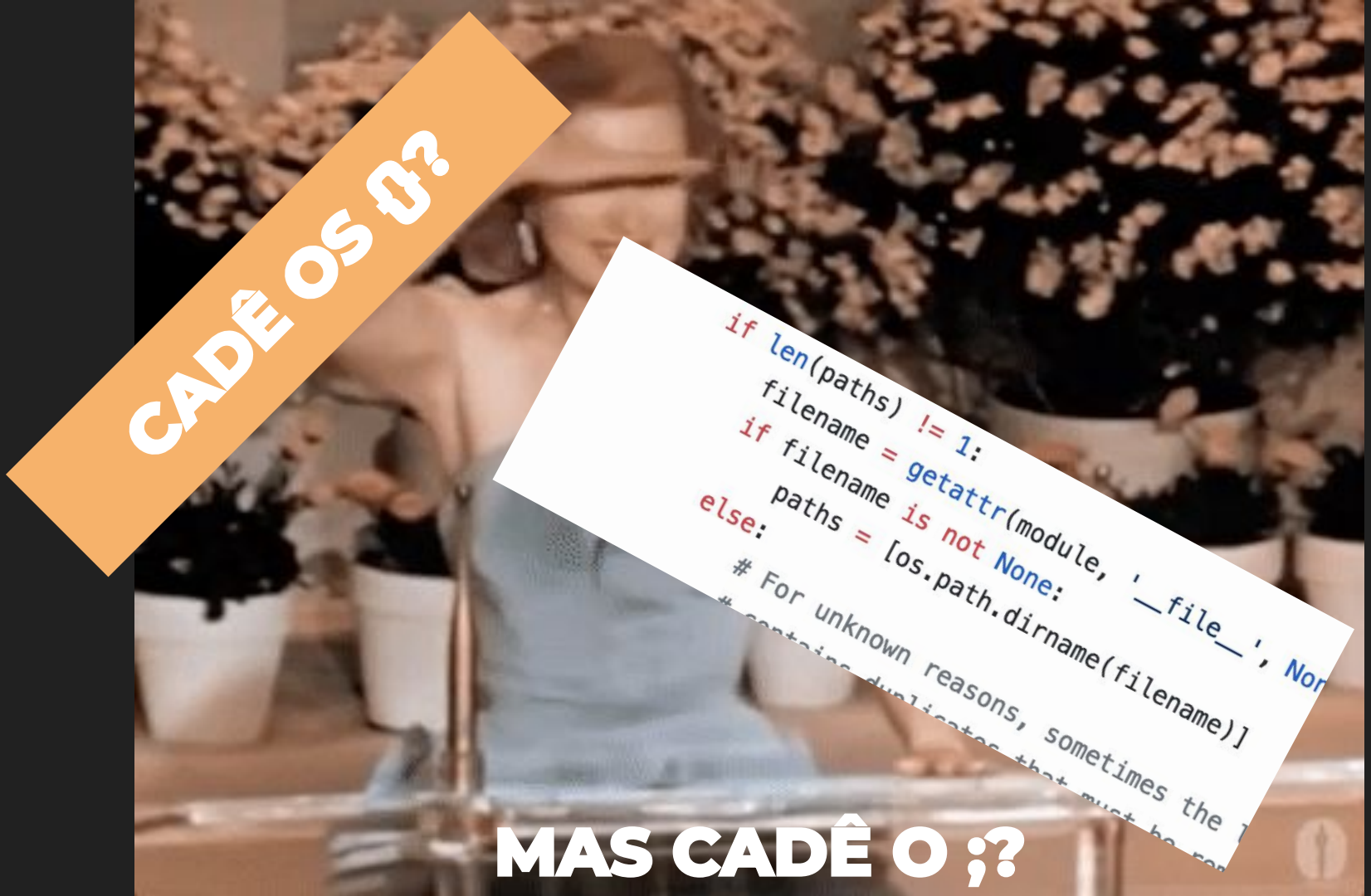
```
def _path_from_module(self, module):
    """Attempt to determine app's filesystem path from its module."""
    # See #21874 for extended discussion of the behavior of this method in
    # various cases.
    # Convert paths to list because Python's _NamespacePath doesn't support
    # indexing.
    paths = list(getattr(module, '__path__', []))
    if len(paths) != 1:
        filename = getattr(module, '__file__', None)
        if filename is not None:
            paths = [os.path.dirname(filename)]
        else:
            # For unknown reasons, sometimes the list returned by __path__
            # contains duplicates that must be removed (#25246).
            paths = list(set(paths))
    if len(paths) > 1:
        raise ImproperlyConfigured(
            "The app module %r has multiple filesystem locations (%r); "
            "you must configure this app with an AppConfig subclass "
            "with a 'path' class attribute." % (module, paths))
    elif not paths:
        raise ImproperlyConfigured(
            "The app module %r has no filesystem location, "
            "you must configure this app with an AppConfig subclass "
            "with a 'path' class attribute." % (module,))
    return paths[0]
```

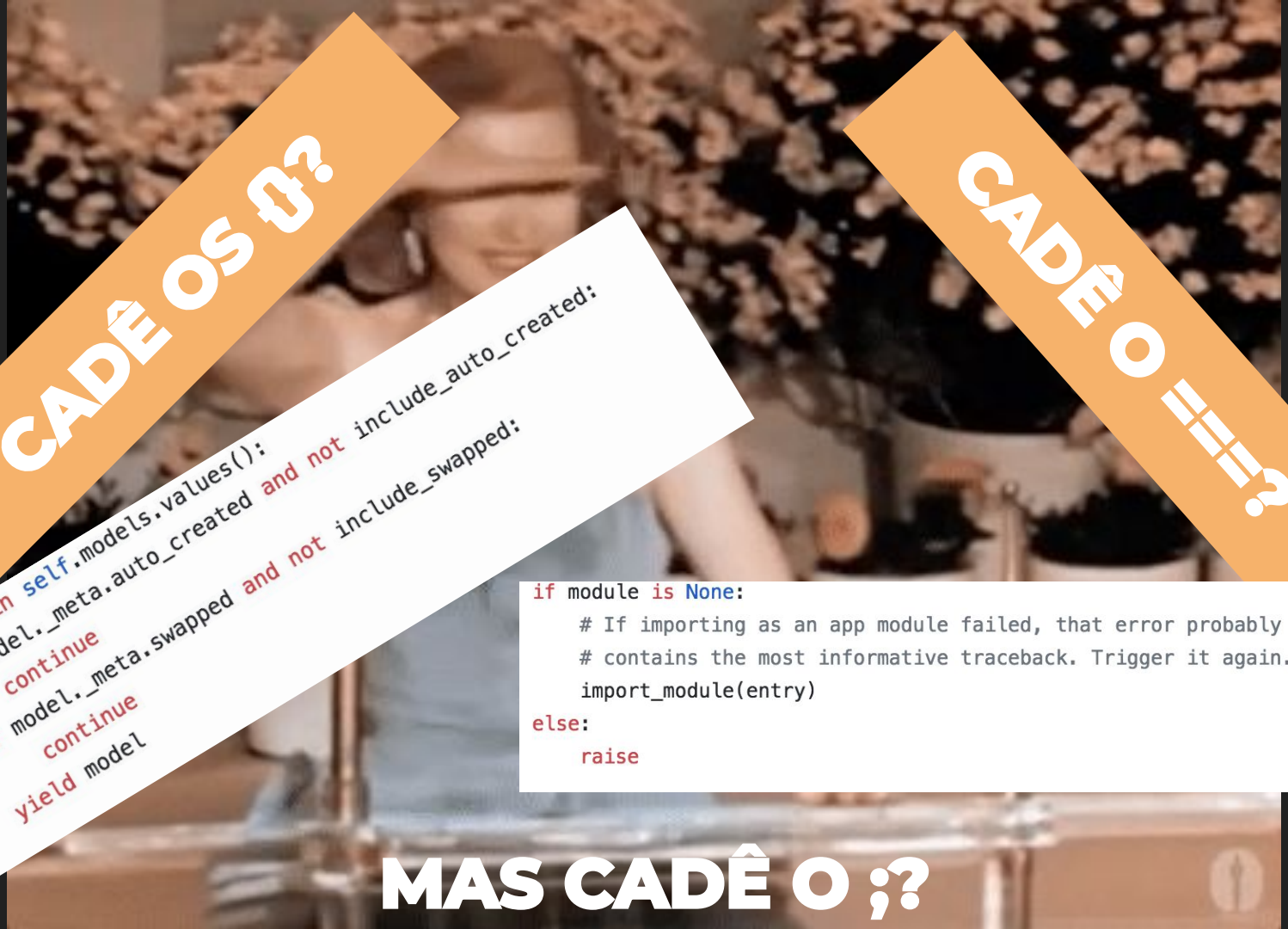
No primeiro código que eu vi!

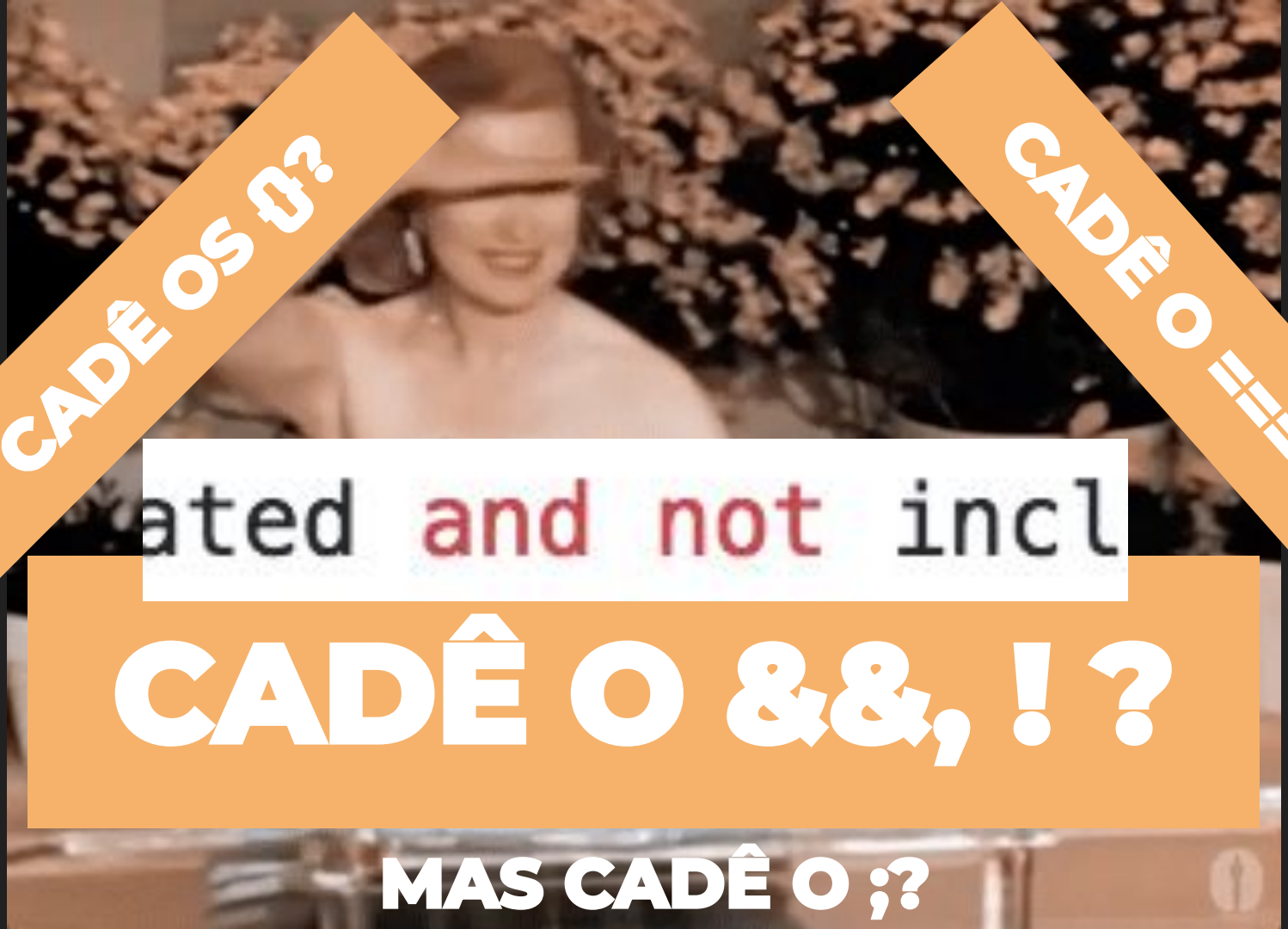
```
def _path_from_module(self, module):
    """Attempt to determine app's filesystem path from its module."""
    # See #21874 for extended discussion of the behavior of this method in
    # various cases.
    # Convert paths to list because Python's _NamespacePath doesn't support
    # indexing.
    paths = list(getattr(module, '__path__', []))
    if len(paths) != 1:
        filename = getattr(module, '__file__', None)
        if filename is not None:
            paths = [os.path.dirname(filename)]
        else:
            # For unknown reasons, sometimes the list returned by __path__
            # contains duplicates that must be removed (#25246).
            paths = list(set(paths))
    if len(paths) > 1:
        raise ImproperlyConfigured(
            "The app module %r has multiple filesystem locations (%r); "
            "you must configure this app with an AppConfig subclass "
            "with a 'path' class attribute." % (module, paths))
    elif not paths:
        raise ImproperlyConfigured(
            "The app module %r has no filesystem location, "
            "you must configure this app with an AppConfig subclass "
            "with a 'path' class attribute." % (module,))
    return paths[0]
```

No primeiro código que eu vi!


filename = getattr(module, '__file__', None)
if filename is not None:
paths = [os.path.dirname(filename)]
else:
# For unknown reasons, sometimes the list returned by __path__
# contains duplicates that must be removed (#25246).
paths = list(set(paths))
if len(paths) > 1:
MAS CADÊ O ;?


CADÊ OS {}?
if len(paths) != 1:
filename = getattr(module, '__file__', Nor
if filename is not None:
paths = [os.path.dirname(filename)]
else:
# For unknown reasons, sometimes the
MAS CADÊ O ;?


CADÊ OS {}?
CADÊ O ==?
for model in self.models.values():
if model._meta.auto_created and not include_auto_created:
continue
if model._meta.swapped and not include_swapped:
continue
yield model
if module is None:
# If importing as an app module failed, that error probably
# contains the most informative traceback. Trigger it again.
import_module(entry)
else:
raise
MAS CADÊ O ;?


CADÊ OS ()?
CADÊ O ===?
ated and not incl
CADÊ O &&, ! ?
MAS CADÊ O ;?


Que tipo de linguagem é essa?

Mas falando sério!

Mas falando sério!

Elas tem suas similaridades!

Começando pelas variáveis!

```
var a = 10;
```


```
a = 10
```

Fonte: https://dev.to/underdogio/python-for-javascript-developers

Depois vem as funções!

MAS CUIDADO

```
def hello:
  world = "Yes! world!"
    print(world) # ← isso é um erro de identação
```

Fonte: https://dev.to/underdogio/pyt

Controles de fluxo

```javascript
if (a > 10) {
  // Alguma coisa
} else if ( a < 5 ) {
  // Outra coisa
} else {
  // O que sobrou
}
```


```python
if a > 10:
  # Alguma coisa
elif a < 5:
  # Outra coisa
else:
  # O que sobrou
```

Fonte: https://dev.to/underdogio/python-for-javascript-developers

Controles de fluxo

```javascript
while (a > 10) {
  a = a + 1
}
```

```python
while a > 10:
  a = a + 1
```

Fonte: https://dev.to/underdogio/python-for-javascript-developers

Controles de fluxo

```javascript
for (let i of n) {
  console.log(i);
}
```

```python
for i in n:
  print(i)
```

Fonte: https://dev.to/underdogio/python-for-javascript-developers

Arrays

```
var array = [1, '2', 9.0 / 3];
```

```
array = [1, '2', 9.0 / 3]
```

Fonte: https://dev.to/underdogio/python-for-javascript-developers

Arrays

```js
array.length;
array.push(4);
array.reverse();
```


```python
len(array)
array.append(4)
array.reverse()
```

Fonte: https://dev.to/underdogio/python-for-javascript-developers

Arrays!!!

```python
array = [1, 2, 3, 4, 5]
array[0:2] # [1, 2, 3]
array[3:] # [4, 5]
array[2, -2] # [3, 4]
```

```javascript
var array = [1, 2, 3, 4, 5];
[ ...array].splice(0, 3); // [1, 2, 3]
[ ...array].splice(3, 2); // [4, 5]
[array[2], array[array.length - 2]] //[3, 4]
```

Fonte: https://dev.to/underdogio/python-for-javascript-developers

Arrays!!!

```
[1, 2] + [3, 4] //"1,23,4" ← WTF???
```


```
[1, 2] + [3, 4] # [1, 2, 3, 4]
```

Fonte: https://dev.to/underdogio/python-for-javascript-developers

Map?, Dictionary?, Object?

You decide!

```
var person = {
  name: `jeff`,
  age: 22
};
```


```
person = {
  'name': 'jeff',
  'age': 22
}
```

Fonte: https://dev.to/underdogio/python-for-javascript-developers

Map?, Dictionary?, Object?

You decide!

```
person.name    //jeff
person['age'] //22

person.gender || 'male'
```

```
person['name'] #jeff
person['age']  #22

person.get('gender', 'male')
```

Fonte: https://dev.to/underdogio/python-for-javascript-developers

None === Null

Fonte: https://dev.to/underdogio/python-for-javascript-developers

Modulos

```js
// module.js
var lyrics = `And I say, hey yeah yeah ... `;

function sing() {
  console.log(lyrics);
}

module.exports = { // ← node
  sing,
  lyrics
};

export { sing, lyrics }; // ← ES6/ES7 ...
```


```python
# module.py

lyrics = "And I say, hey yeah yeah ... "

def sing():
  print(lyrics)
```

Fonte: https://dev.to/underdogio/python-for-javascript-developers

Modulos

```
const module = require('./module');
import module from 'module';

module.sing()
```

```
import module


module.sing()
```

Fonte: https://dev.to/underdogio/python-for-javascript-developers

Modulos

```
const sing = require('./module').sing;
import { sing } from 'module';

sing();
```

```
from module import sing

sing()
```

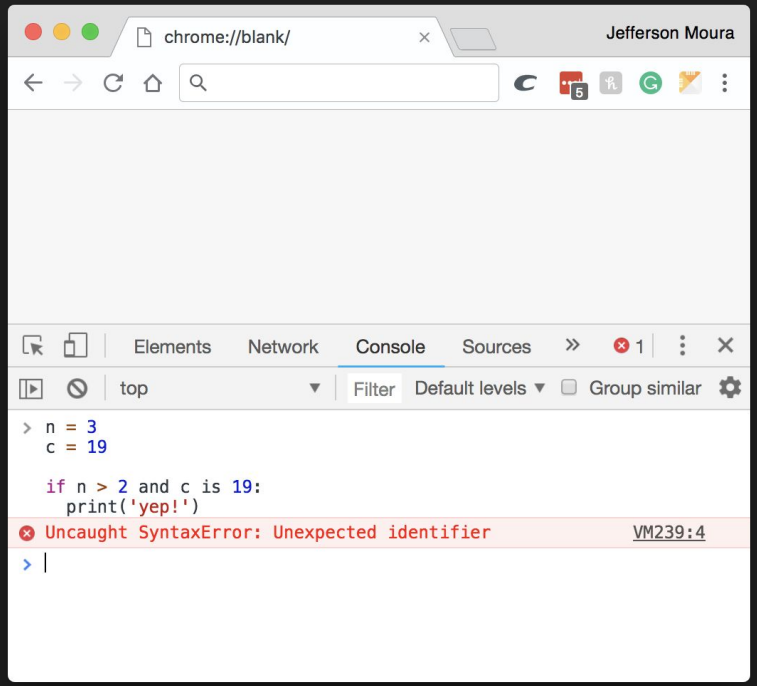
Fonte: https://dev.to/underdogio/python-for-javascript-developers

Mas Apesar das diferenças!

Todo mundo se respeita!

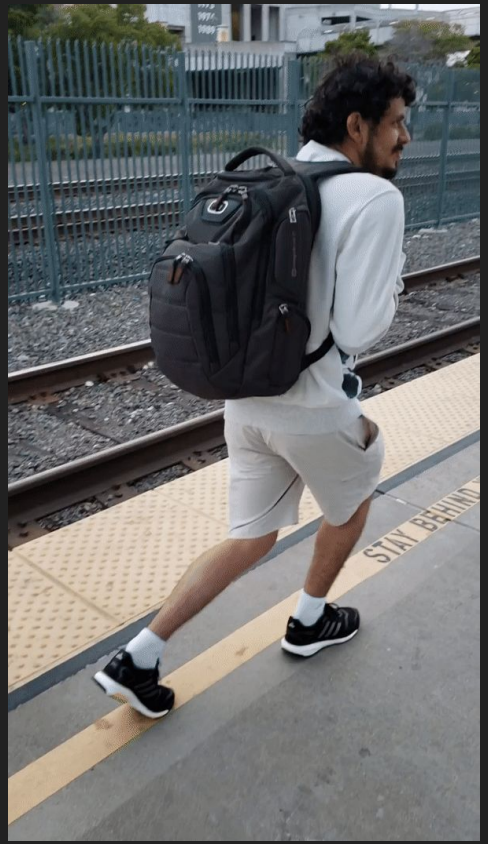
E ambos funcionam bem em todos os lugares!

E ambos funcionam bem em todos os lugares!



Onde são suportados! Claro!

Então, é só fazer a melhor escolha e ser feliz!




<- Slides Disponíveis aqui!
Perguntas??