

# Python: Um Workshop Orientado a Objectos

João Rebelo Pires

NUCC-FCUP (DCC)

18 de Fevereiro de 2018

# Conteúdos

- História
- Programas Simples
- Classes

# História

- Criado por Guido van Rossum;

# História

- Criado por Guido van Rossum;
- Lançado pela primeira vez em 1991;

# História

- Criado por Guido van Rossum;
- Lançado pela primeira vez em 1991;
- Nome é referência aos Monty Python;

# História

- Criado por Guido van Rossum;
- Lançado pela primeira vez em 1991;
- Nome é referência aos Monty Python;
- Linguagem multi-paradigma;

# História

- Criado por Guido van Rossum;
- Lançado pela primeira vez em 1991;
- Nome é referência aos Monty Python;
- Linguagem multi-paradigma;
- Fornece construtores que permitem programação "limpa" a qualquer escala.

# Hello World (típico...)

Tão simples como:

```
print "Hello World"
```



## Somar dois números introduzidos pelo utilizador

```
x1 = int(input("Introduz o primeiro inteiro: "))  
x2 = int(input("Introduz o segundo inteiro: "))  
print (x1 + x2)
```

# Usando listas

```
lst = []  
for i in range(1, 21):  
    lst.append(i)  
print lst  
for item in lst:  
    print item
```

## Prints com formatação

```
sum_x = 0
for x in range(1, 11):
    acc = x + x ** 2 + x ** 3
    sum_x += x
    print('{0:2d} {1:3d} {2:4d} {3:4d}'.format(x, x *
        ↪ x, x * x * x, acc))

print "Soma de 1 a 10: %d\nUltimo valor em acc: %d"
    ↪ %(sum_x, acc)
```

# Questões?

# Introdução

- Objectos;

# Introdução

- Objectos;
- Instâncias;

# Introdução

- Objectos;
- Instâncias;
- Métodos;

# Introdução

- Objectos;
- Instâncias;
- Métodos;
- Atributos;



# Introdução

- Objectos;
- Instâncias;
- Métodos;
- Atributos;
- Herança.

# Métodos especiais para objectos

- `__init__()`

# Métodos especiais para objectos

- `__init__()`
- `__del__()`

# Métodos especiais para objectos

- `__init__()`
- `__del__()`
- `__repr__()`

# Métodos especiais para objectos

- `__init__()`
- `__del__()`
- `__repr__()`
- `__str__()`

# Métodos especiais para objectos

- `__init__()`
- `__del__()`
- `__repr__()`
- `__str__()`
- `__eq__()`

# Métodos especiais para objectos

- `__init__()`
- `__del__()`
- `__repr__()`
- `__str__()`
- `__eq__()`
- `__ne__()`

# Métodos especiais para objectos

- `__init__()`
- `__del__()`
- `__repr__()`
- `__str__()`
- `__eq__()`
- `__ne__()`
- `__hash__()`



## Exemplo - class Trick

```
class Trick(object):  
    def __init__(self, name, score):  
        self.name = name  
        self.score = score  
  
    def __repr__(self):  
        return "Trick({0:>s})".format(self.__str__())  
  
    def __str__(self):  
        return str(self.name) + ", " + str(self.score)
```

## Exemplo - class Trick

```
def __eq__(self, other):  
    if self.name == other.name and self.score ==  
        ↪ other.score:  
        return True  
    return False  
  
def __ne__(self, other):  
    return not self.__eq__(other)  
  
def __hash__(self):  
    return hash(repr(self))
```

## Exemplo - class Dog

```
class Dog(object):
    def __init__(self, kind, name=None,
        ↪ chipnumber=None):
        self.kind = kind
        self.name = name
        self.tricks = set([])
        self._chipnumber = chipnumber

    def __repr__(self):
        return "Dog({0:>s})".format(self.__str__())
```

## Exemplo - class Dog

```
def __str__(self):  
    if self.name is not None:  
        return str(self.kind) + ", " +  
            ↪ str(self.name)  
    return str(self.kind)  
  
def __eq__(self, other):  
    if self._chipnumber is not None and  
        ↪ other._chipnumber is not None:  
        return self._chipnumber ==  
            ↪ other._chipnumber  
    return False
```

## Exemplo - class Dog

```
def __ne__(self, other):  
    return not self.__eq__(other)  
  
def __hash__(self):  
    return hash(repr(self))  
  
def addchipnumber(self, chipnumber):  
    if self._chipnumber is not None:  
        raise ValueError("Dog already has a  
            ↪ chipnumber")  
    else:  
        self._chipnumber = chipnumber
```

## Exemplo - class Dog

```
def addtrick(self, trick):  
    self.tricks.add(trick)  
  
def gettricks(self):  
    t = []  
    for trick in list(self.tricks):  
        t.append(trick.name)  
    return set(t)
```

## Exemplo - class Competition

```
class Competition(object):  
    def __init__(self):  
        self.participants = {}  
        self._table = {1: None, 2: None, 3: None}  
        self._tricks = {}  
  
    def __repr__(self):  
        return  
        ↪ "Competition({0:>s})".format(self.__str__())
```

## Exemplo - class Competition

```
def __str__(self):  
    ans = {}  
    for set_key in self.participants:  
        ans[set_key] =  
            ↪ list(self.participants[set_key])  
  
    return str(ans)  
  
def addparticipant(self, participant):  
    if participant._chipnumber is None:  
        raise AttributeError("Participant does  
            ↪ not have a chipnumber!")
```



## Exemplo - class Competition

```
elif participant.kind not in
    ↪ self.participants:
        self.participants[participant.kind] =
            ↪ {participant}
elif participant in
    ↪ self.participants[participant.kind]:
        raise ValueError("Participant already
            ↪ competing!")
else:
    self.participants[participant.kind].\
        add(participant)
```

## Exemplo - class Competition

```
def podium(self):
    if self._table[1] is None:
        print "No winner yet"
    elif self._table[2] is None:
        print "First Place: {} with {}
        ↪ point(s)".format(self._table[1][0].name,
        ↪ self._table[1][1])
    elif self._table[3] is None:
        print "First Place: {} with {}
        ↪ point(s)".format(self._table[1][0].name,
        ↪ self._table[1][1])
    print "Second Place: {} with {}
    ↪ point(s)".format(self._table[2][0].name,
    ↪ self._table[2][1])
```

## Exemplo - class Competition

else:

```
print "First Place: {} with {}  
    ↪ point(s)".format(self._table[1][0].name,  
    ↪ self._table[1][1])  
print "Second Place: {} with {}  
    ↪ point(s)".format(self._table[2][0].name,  
    ↪ self._table[2][1])  
print "Third Place: {} with {}  
    ↪ point(s)".format(self._table[3][0].name,  
    ↪ self._table[3][1])
```

## Exemplo - class Competition

```
def addtrick(self, trick):  
    if trick.name in self._tricks:  
        raise ValueError("This is already a trick  
        ↪ for this competition!")  
    else:  
        self._tricks[trick.name] = trick.score
```

## Exemplo - class Competition

```
def simulate(self, debug=False):
    tablelist = []
    for kind in self.participants:
        for dog in self.participants[kind]:
            score = 0
            tricks = dog.gettricks()
            for trick in tricks:
                if trick in self._tricks:
                    score += self._tricks[trick]
            tablelist.append((dog, score))
```

## Exemplo - class Competition

```
sorted_table = sorted(tablelist,  
    ↪ key=operator.itemgetter(1), reverse=True)  
it = 1  
for (dog, score) in sorted_table:  
    self._table[it] = (dog, score)  
    it += 1  
if debug:  
    for (dog, _) in sorted_table:  
        print dog.name  
        it += 1
```

## Exemplo - Utilização

```
t1 = Trick("roll over", 1)
```

## Exemplo - Utilização

```
t1 = Trick("roll over", 1)
rufus = Dog("golden retriever", "rufus")
```



## Exemplo - Utilização

```
t1 = Trick("roll over", 1)
rufus = Dog("golden retriever", "rufus")
rufus.addchipnumber(1001001)
```

## Exemplo - Utilização

```
t1 = Trick("roll over", 1)
rufus = Dog("golden retriever", "rufus")
rufus.addchipnumber(1001001)
comp = Competition()
```

## Exemplo - Utilização

```
t1 = Trick("roll over", 1)
rufus = Dog("golden retriever", "rufus")
rufus.addchipnumber(1001001)
comp = Competition()
comp.addparticipant(rufus)
```

## Exemplo - Utilização

```
t1 = Trick("roll over", 1)
rufus = Dog("golden retriever", "rufus")
rufus.addchipnumber(1001001)
comp = Competition()
comp.addparticipant(rufus)
comp.addtrick(t1)
```

## Exemplo - Utilização

```
t1 = Trick("roll over", 1)
rufus = Dog("golden retriever", "rufus")
rufus.addchipnumber(1001001)
comp = Competition()
comp.addparticipant(rufus)
comp.addtrick(t1)
comp.simulate()
```

## Exemplo - Utilização

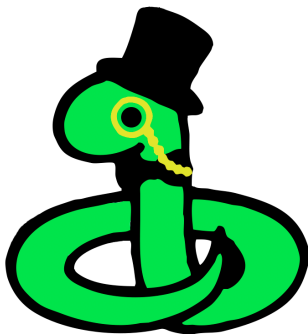
```
>>> comp.podium()
```

## Exemplo - Utilização

```
>>> comp.podium()
```

First Place: rufus with 1 point(s)

Vai um live coding?



 jrbartowski

 joaorgpires@gmail.com