



Functools

Live de python # 136

SOBRE O CANAL



O canal é mantido por uma campanha financiamento coletivo

<https://apoia.se/livedepython>

ou

<https://picpay.me/dunossauro>

Aciássio Araújo, Ademar Peixoto, Adão Oliveira, Alan Soder, Alberto Tavares, Alessandro Lopes, Alex Lima, Alexandre Tsuno, Alexandre Villares, Alexandre Santos, Alexandre Sá, Alexandre Harano, Alysson Oliveira, Amanda Magalhães, Amaziles Carvalho, Andre Rodrigues, Antonio Ribeiro, Bernardo Fontes, Bianca Rosa, Bruno Rocha, Bruno Oliveira, Caio Nascimento, Carlos Cardoso, Carlos Santos, Cleiton Souza, Davi Ramos, Davi Lima, Diego Guimarães, Dilenon Delfino, Douglas Bastos, Edson Braga, Eduardo Nunes, Eduardo Pazzi, Elias Soares, Emerson Lara, Eugenio Mazzini, Everton Alves, Fabrício Coelho, Fernando Lanfranchi, Fernando Vier, Filipe Cruz, Flavkaze , Flávia Batista, Franklin Silva, Fábio Serrão, Gabriel Simonetto, Gabriela Santiago, Gabrielly Andrade, Geandreson Costa, Gladson Menezes, Hélio Neto, Isaac Ferreira, Israel Fabiano, Ivan Vicente, Johnny Tardin, João Lugão, Jonatas Leon, Jonatas Oliveira, Jonatas Oliveira, Jones Leite, Jones Lourenço, Jones Leite, Jorge Plautz, José Prado, Jovan Costa, Juan Gutierrez, Leonardo Cruz, Leonardo Galani, Lucas Mendes, Lucas Valino, Lucas Neris, Lucas Polo, Luciano Ratamero, Luiz Lima, Luiz Bruno, Maiquel Leonel, Marcela Campos, Marcello Benigno, Marcos Jesus, Maria Clara, Mateus Braga, Melisa Campagnaro, Otavio Carneiro, Patric Lacouth, Patrick Gomes, Paulo Tadei, Paulo Gonçalves, Rafael Dias, Rafael Peixoto, Reinaldo Silva, Renne Rocha, Ricardo Schalch, Rodrigo Vaccari, Rodrigo Ferreira, Régis Tomkiel, Samanta Cicilia, Thiago Araujo, Thiago Bueno, Tiago Cordeiro, Tyrone Damasceno, Valdir Junior, Vicente Marcal, Victor Orsolan, Victor Geraldo, Vinícius Ferreira, Vinícius Bastos, Vítor Zanoni, William Oliveira, Willian Lopes, Willian Rosa, Willian Lopes, Wladimir Falcão,



OBRIGADO A VOCÊ



Sobre o que não vamos falar hoje?



- Imutabilidade (pyrsistent)
- Avaliação preguiçosa
- Iteradores (itertools)
- Tipos algébricos (pymonad, oslash)
- Efeitos colaterais (effect)
- QuickTest (hypotesis)



0. O módulo functools

Uma breve revisão

1. Funções parciais

Aumentando a legibilidade em HOFs

2. LRU Cache

Melhorando performance em chamadas

3. Single dispatch

Simplificando grandes estruturas em funções

5. Reduce

Concatenando coisas sem loops

6. etc...

O que mais der tempo

Func tools

Uma breve explicação

O módulo Functools



O módulo `functools` é para funções de ordem superior: funções que atuam ou retornam outras funções. Em geral, qualquer objeto que pode ser chamado pode ser tratado como uma função para os fins deste módulo.

`functools` — Higher-order functions and operations on callable objects

Source code: `Lib/functools.py`

The `functools` module is for higher-order functions: functions that act on or return other functions. In general, any callable object can be treated as a function for the purposes of this module.

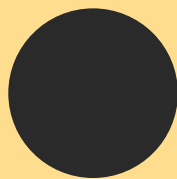
Funções como objeto, HOFs e closures



Temos uma mini série só sobre o assunto e uma espécie de “livro” sobre programação funcional



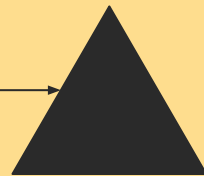
Um básico necessário (Funções)



Entrada



Função



Saída

Um básico necessário (HOFs)



Entrada
(Função)



Função

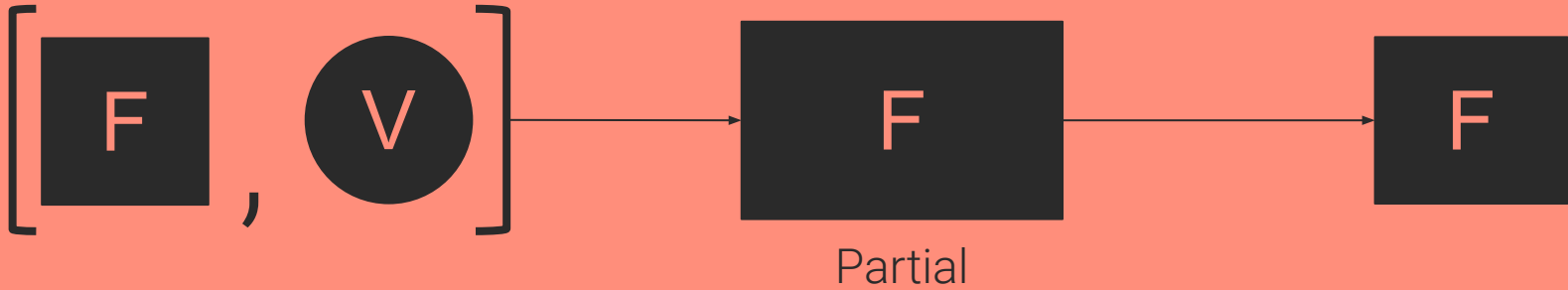


Saída
(Função)

Simplificando
chamadas

parti
al

Partial



Um exemplo de código com partial



```
1  from functools import partial
2
3
4  def exponenciação(x, y):
5      return x ** y
6
7
8  quadrado = partial(exponenciação, 2)
9  cubo = partial(exponenciação, 3)
10 quarta = partial(exponenciação, 4)
```

Partial



A principal ideia do partial é gerar novas funções “simplificada” a nível de quantidade de argumentos

```
soma(x, y)
```



```
soma_2 = partial(soma, 2)
```

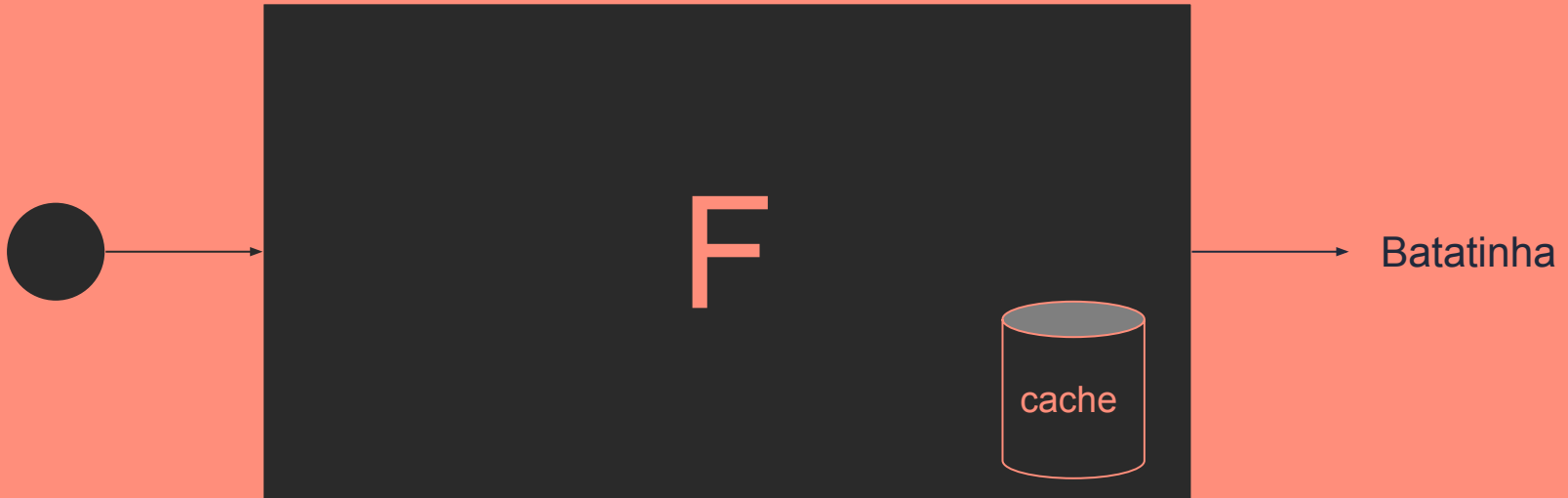


```
soma_2(2) == 4
```

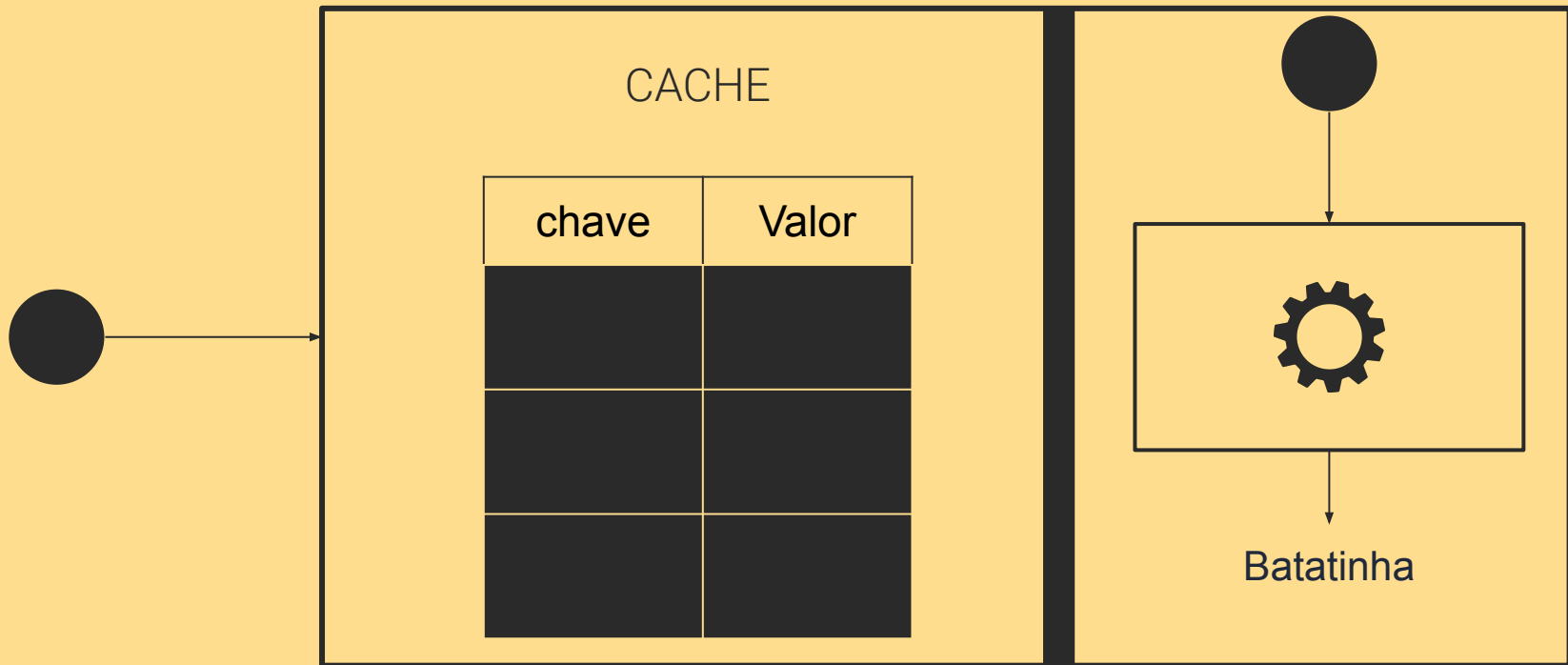
Armazenando
chamadas
anteriores

lru_
cache

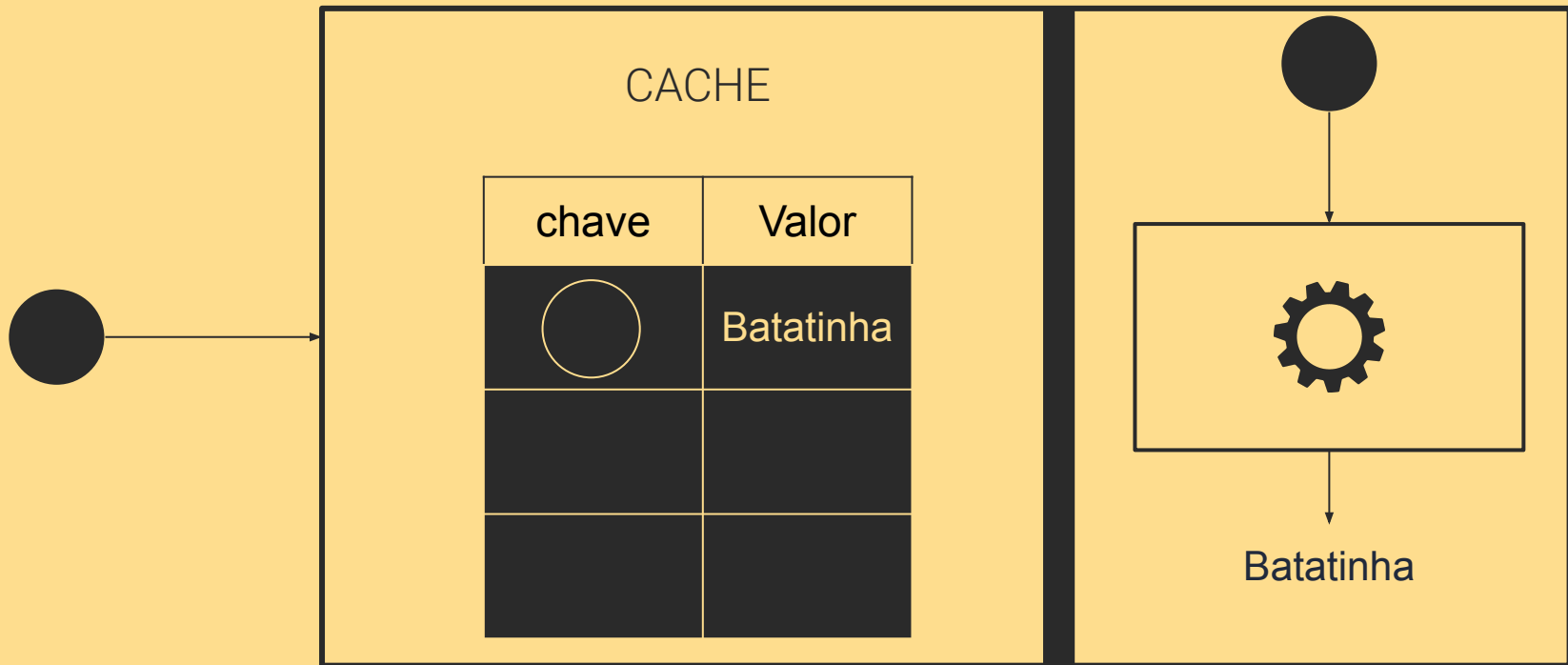
lru_cache



lru_cache




lru_cache

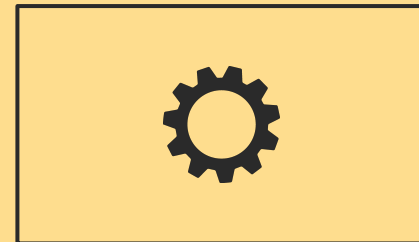


lru_cache



CACHE

chave	Valor
	Batatinha

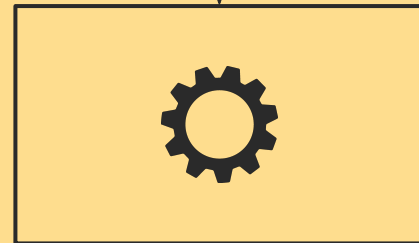


lru_cache



CACHE

chave	Valor
	Frita
	Batatinha



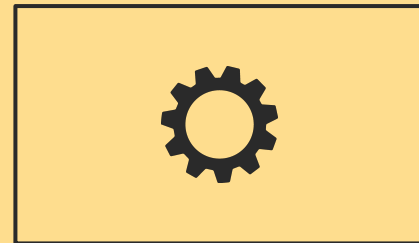
Frita

lru_cache



CACHE

chave	Valor
	Batatinha
	Frita



Um exemplo de código com lru_cache



```
1  from functools import lru_cache
2  from time import sleep
3
4  @lru_cache(maxsize=2)
5  def id_delay(x):
6      sleep(3)
7      return x
8
9  id_delay(1) # demora 3 segundos
10 id_delay(3) # demora 3 segundos
11 id_delay(1) # Usa o cache
12 id_delay(3) # Usa o cache
```

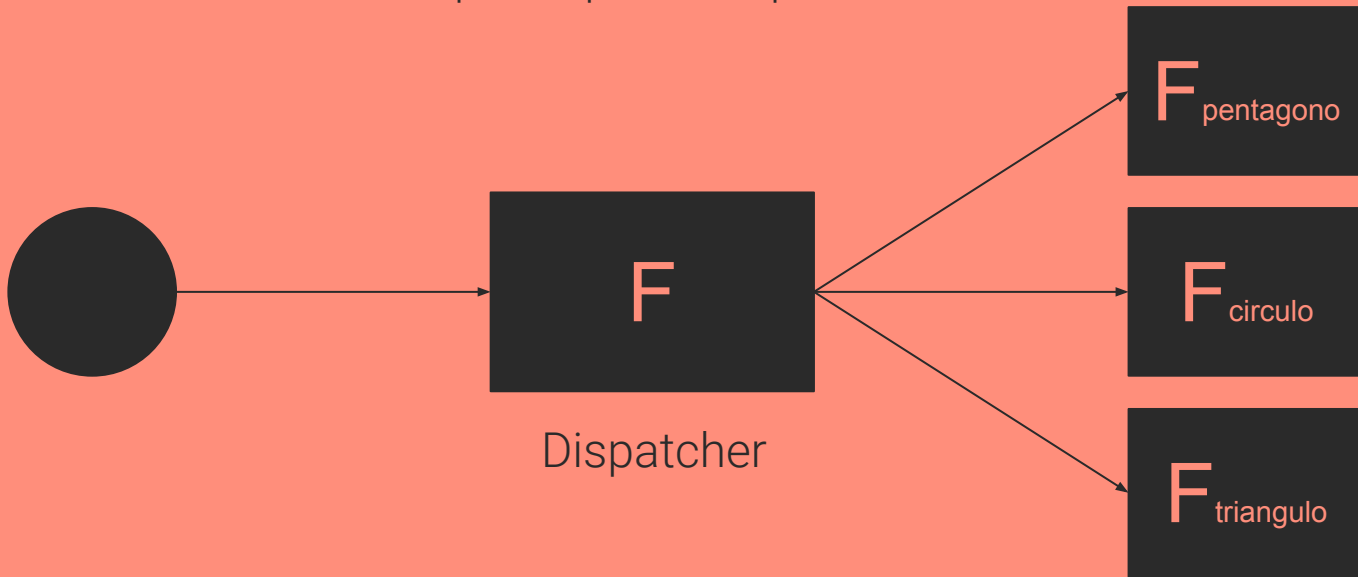
Resolvendo tipos
diferentes com
funções
diferentes

single
dispatch

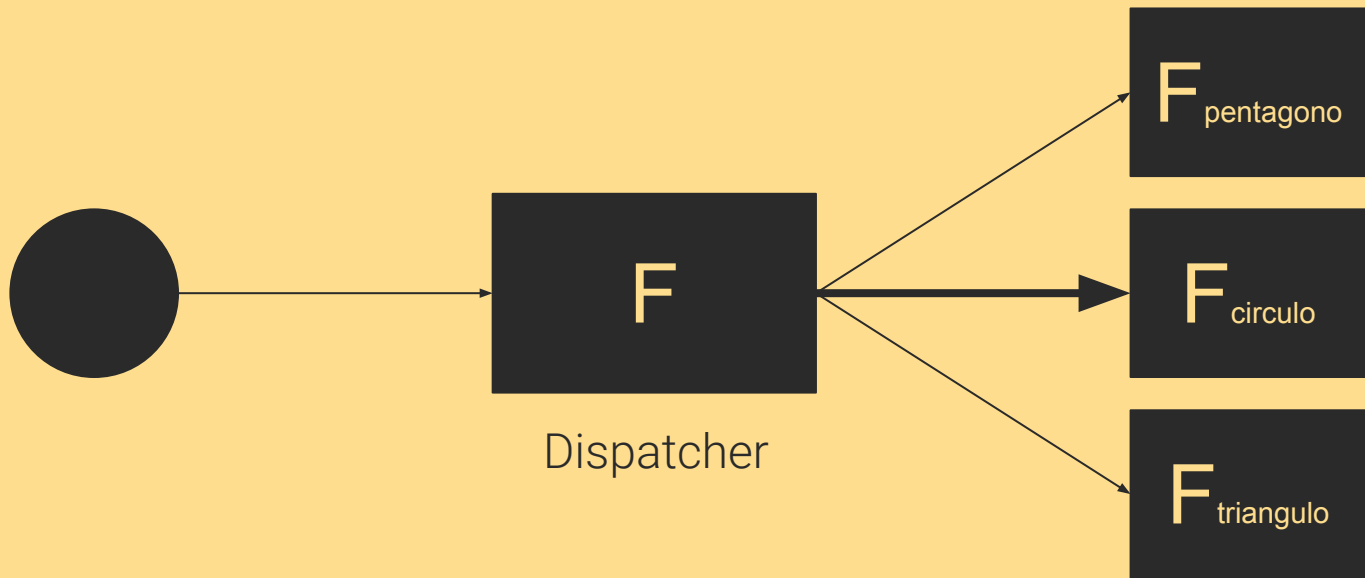
Single Dispatch



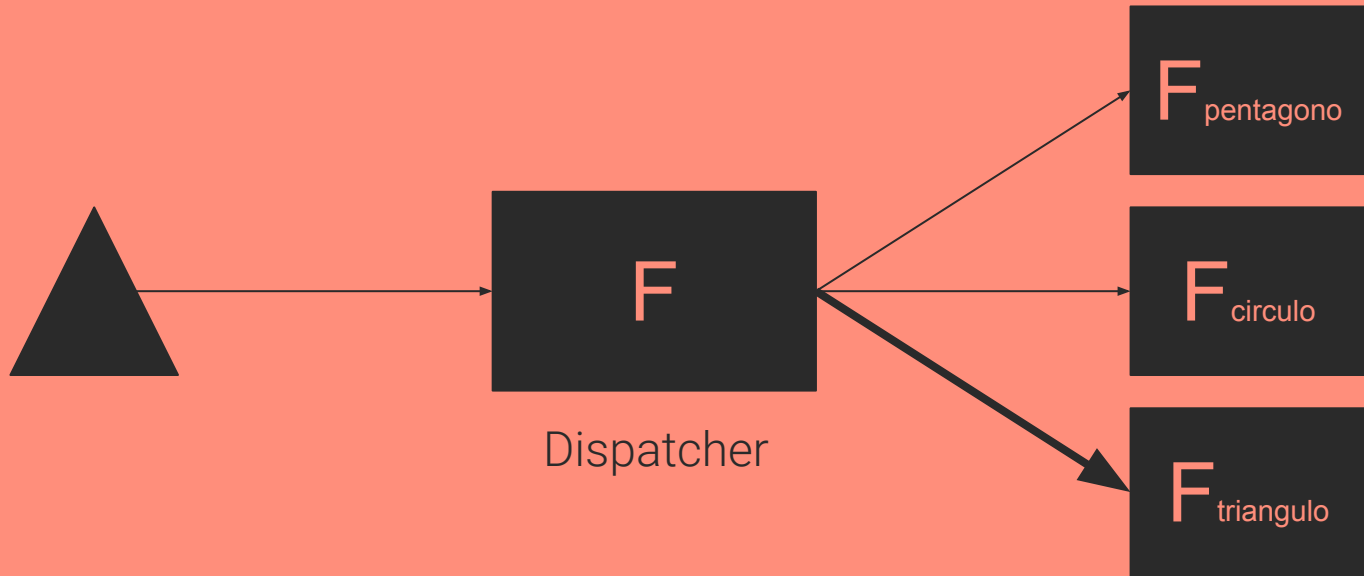
Single dispatch é uma maneira de simplificar fluxos em chamadas de funções baseando-se no tipo do primeiro parâmetro.



Single Dispatch



Single Dispatch



redu
ce

Evitando loops
para concatenar

SOBRE O CANAL



O canal é mantido por uma campanha financiamento coletivo

<https://apoia.se/livedepython>

ou

<https://picpay.me/dunossauro>