Python 3.10: Welcome to pattern matching

@laysauchoa

self.intro()

Laysa Uchoa

@laysauchoa

github.com/laysauchoa

- backend developer at e-mobility (Munich)
- dabbling in Python for 4+ years
- Python communities <3
- dystopian books, tea and carnaval

Structure

Part 1: Python 3.10

Part 2: Pattern Matching 

Part 3: Final thoughts

@laysauchoa

Python 3.10

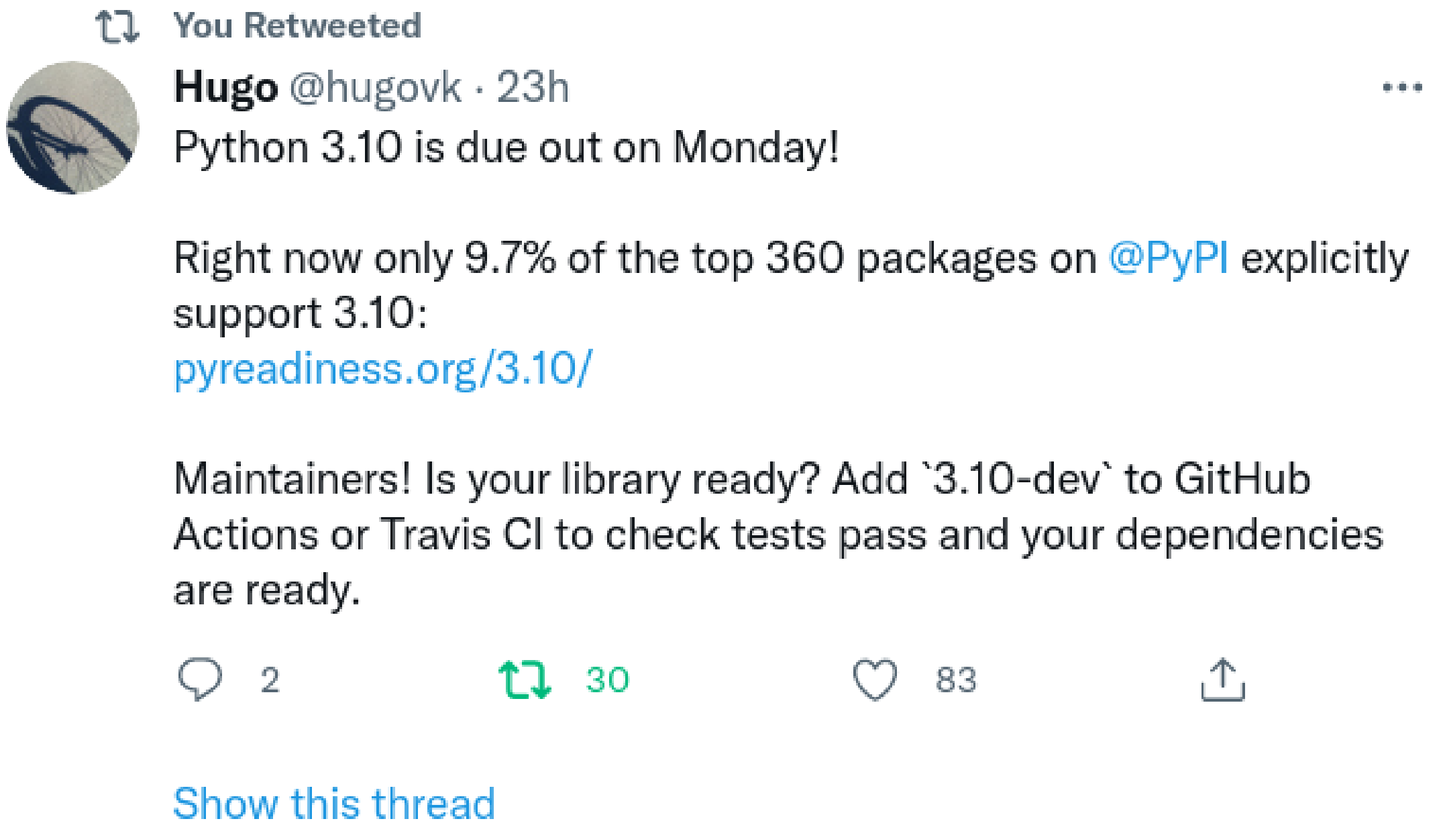
Meaningful messages


```
/Users/anthonyshaw/projects/cpython-3.10/python.exe -X dev Tests/run_libregr_tests.py test_buffer -v
Traceback (most recent call last):
  File "/Users/anthonyshaw/CLionProjects/pyjion/Tests/run_libregr_tests.py", line 8, in <module>
    pyjion.enable_graph()
AttributeError: module 'pyjion' has no attribute 'enable_graph'. Did you mean: 'enable_graphs'?
```

@laysauchoa

Python 3.10

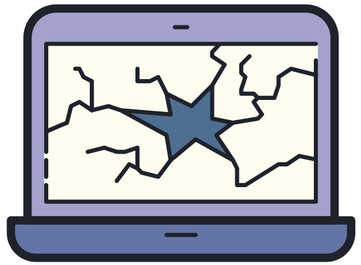
OSS



@laysauchoa

Python 3.10

Breaking Changes



Leah Neukirchen @LeahNeukirchen · Sep 27

You won't believe how much software breaks because **Python 3.10** has a two-digit minor version.

51 194 1.3K

Show this thread

Download the latest source release

Download Python 3.9.7

Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

@laysauchoa

Python 3.10

Party

@laysauchoa

https://github.com/python/cpython/blob/main/Doc/whatsnew/3.10.rst

@laysauchoa

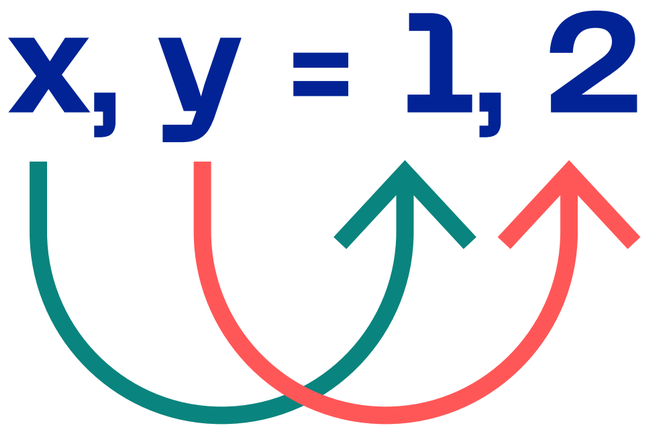
Python 3.10

What is new?

@laysauchoa

Primitive

Pattern matching



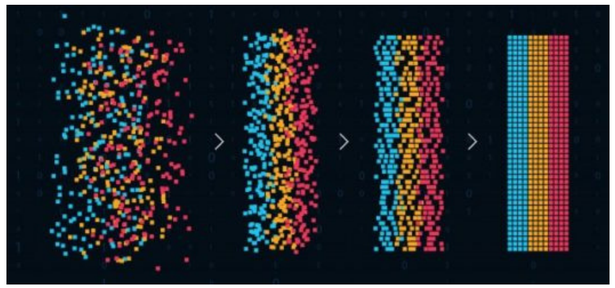
Primitive

Pattern matching

```
try:
    # do something that may fail
except ILikeYouException:
    # do something about it
except YouAreBeingExceptional:
    # do something about it
```

Pattern matching

"either it will or will not be a match"



not a dating app
not a pattern recognition

Game

Pattern matching

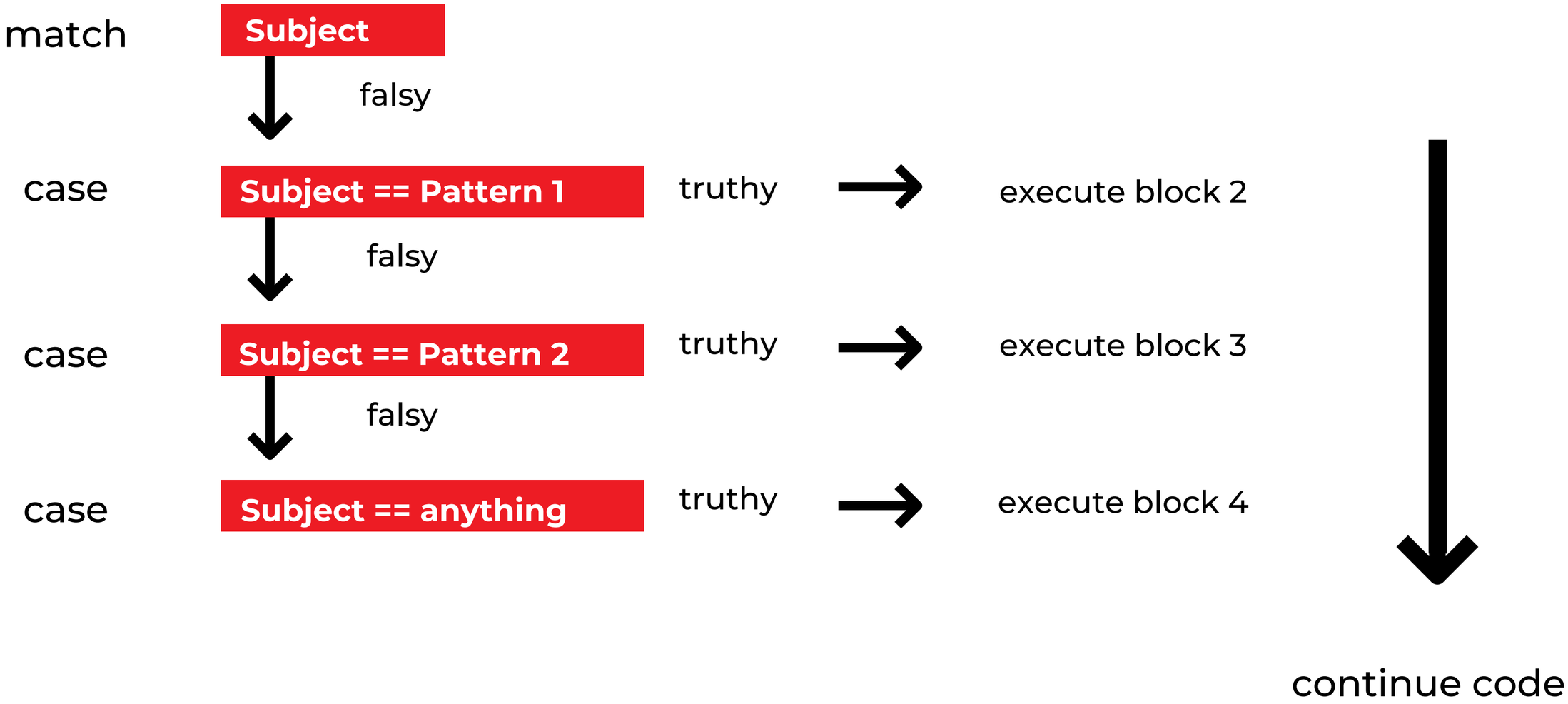
or

First Game

Pattern matching

```
In [1]: form = "triangle"
   ...:
   ...: match form:
   ...:     case "circle":
   ...:         print("Circle: ⚪")
   ...:     case "square":
   ...:         print("Square: ⬜")
   ...:     case "triangle":
   ...:         print("Triangle: 🔺")
   ...:     case _:
   ...:         print("Any other form!")
   ...:
```

Pattern matching



Don'ts

Pattern matching

```
In [1]: form = "triangle"
   ...:
   ...: match form:
   ...:     case _:
   ...:         print("Any other form!")
   ...:     case "circle":
   ...:         print("Circle: ⚪")
   ...:     case "square":
   ...:         print("Square: ⬜")
   ...:     case "triangle":
   ...:         print("Triangle: 🔺")
```

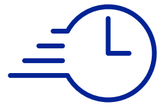
With and Without

Pattern matching

```python
def store_elements(elements: List[str])-> str:
    match elements:
        case ["triangle"]:
            print("triangle is stored")
        case ["triangle", obj]:
            print(f"triangle and {obj} are stored")
        case ["circle", *objects]:
            print(f"circle stored with: {objects}")
        case _:
            print("no matches found")
```

```python
def store_elements(elements: List[str]) -> str:
    if elements == ["triangle"]:
        print("triangle is stored")
    elif len(elements) == 2 and elements[0] == "triangle":
        obj = elements[1]
        print(f"triangle and {obj} are stored")
    elif len(elements) >=2 and elements[0] == "circle":
        obj = elements[1:]
        print(f"circle stored with: {objects}")
    else:
        print("no matches found")
```

%timeit

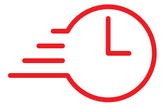


Pattern matching

```
In [46]: %timeit store_elements_with_match(["circle", "triangle","square"])
367 ns ± 5.11 ns per loop (mean ± std. dev. of 7 runs, 1000000 loops each)

In [47]: %timeit store_elements_without_match(["circle", "triangle","square"])
380 ns ± 5.39 ns per loop (mean ± std. dev. of 7 runs, 1000000 loops each)
```

%timeit

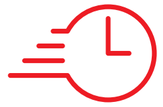


Pattern matching

```
In [48]: %timeit store_elements_without_match([])
239 ns ± 12 ns per loop (mean ± std. dev. of 7 runs, 1000000 loops each)

In [49]: %timeit store_elements_with_match([])
177 ns ± 9.92 ns per loop (mean ± std. dev. of 7 runs, 10000000 loops each)
```

%timeit



Pattern matching

```
# Python 3.10
In [48]: %timeit store_elements_without_match([])
239 ns ± 12 ns per loop (mean ± std. dev. of 7 runs, 1000000 loops each)

# Python 3.7
In [3]: %timeit store_elements_without_match([])
272 ns ± 18.7 ns per loop (mean ± std. dev. of 7 runs, 1000000 loops each)
```

With and Without

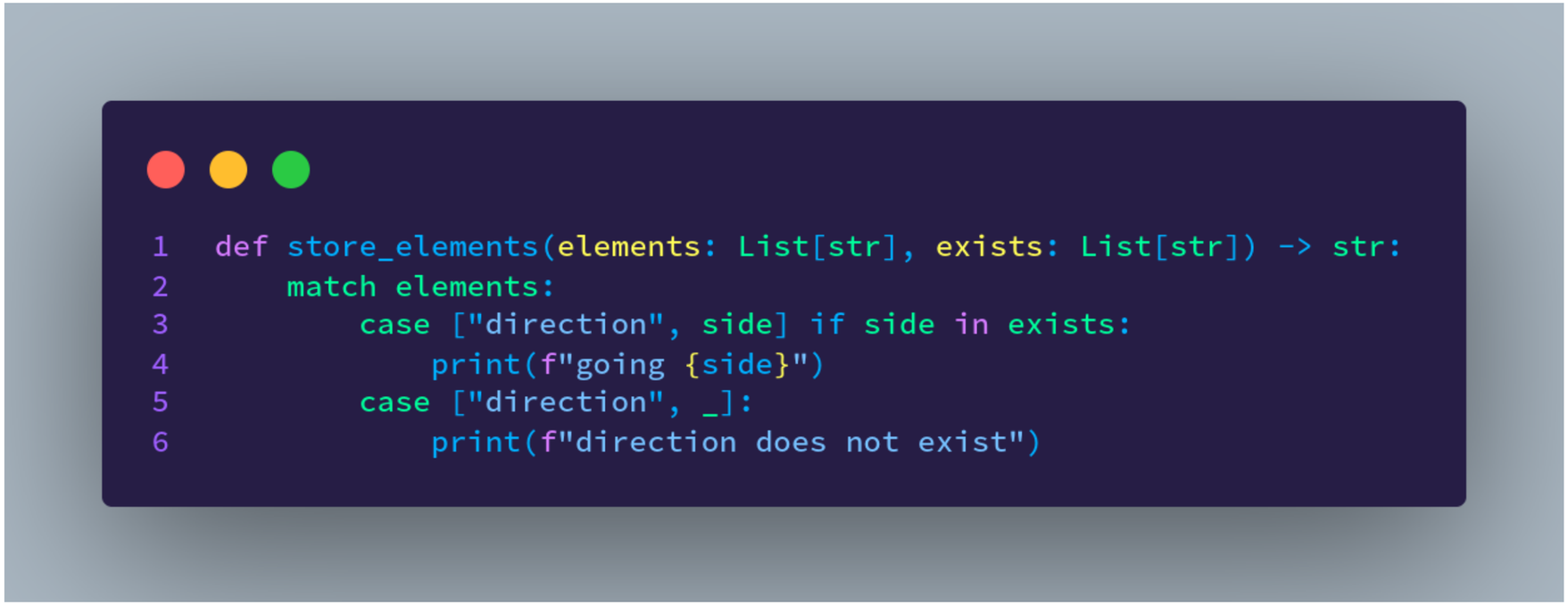
Pattern matching

```python
def count_elements(number: int) -> str:
    match number:
        case 0:
            print("No elements!")
        case 1:
            print("One element!")
        case 2:
            print("Two elements!")
        case _:
            print("wildcard!")
```

```python
def count_elements(number: int) -> str:
    if number == 0:
        print("No elements!")
    elif number == 1:
        print("One element!")
    elif number == 2:
        print("Two elements!")
    else:
        print("wildcard!")
```

Guard

Pattern matching



```python
def store_elements(elements: List[str], exists: List[str]) -> str:
    match elements:
        case ["direction", side] if side in exists:
            print(f"going {side}")
        case ["direction", _]:
            print(f"direction does not exist")
```

Composable Patterns

Pattern matching

top

left

right

bottom

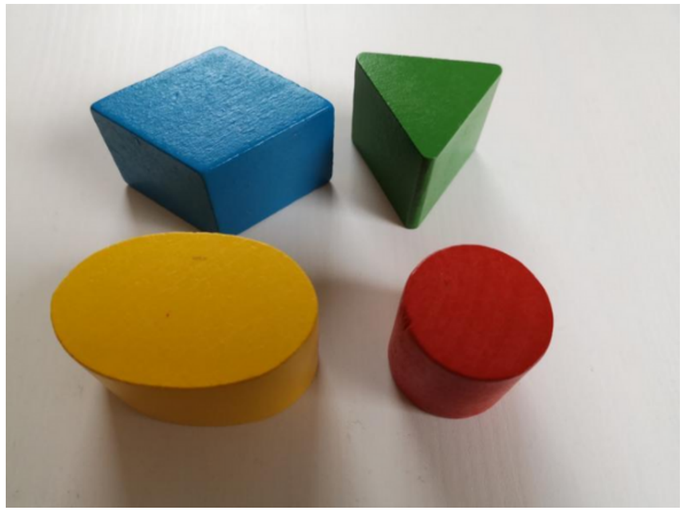
Composable Patterns

Pattern matching

```python
def store_elements(elements: List[str]) -> str:
    match elements:
        case ["direction", ("bottom" | "top" | "left" | "right") as side]:
            print(f"The side is {side}")
```

Matching Objects

Pattern matching



```python
@dataclass
class Circle:
    color: str
    radius: float

element = Circle("pink", 10.0)

match element:
    case Circle(color="red", radius=value):
        print(f"Red circle stored with radius={value}")
    case Circle(color=element_color, radius=value) if value >= 10.0:
        print(f"Circle with radius={value}")
    case _:
        print("no match found.")
```

>>> Circle with radius=10.0

Dictionary

Pattern matching

Python 3.10

Dict Pattern matching

laysa 🏹 @laysauchoa · 2h

Python 3.10.0rc1: help🤦, why is this happening?

```
>>> event = {"size":3}
>>> match event:
...     case {}:
...         print("Empty dict!")
...
Empty dict!

>>> event = (1, 2, 3)
>>> match event:
...     case ():
...         print("Empty tuple!")
...
>>>
```

@laysauchoa

Thoughts

Pattern matching

Questions

If you want to know anything more about my work in communities or share your ideas, **feel free to talk to me.**

Thank you!