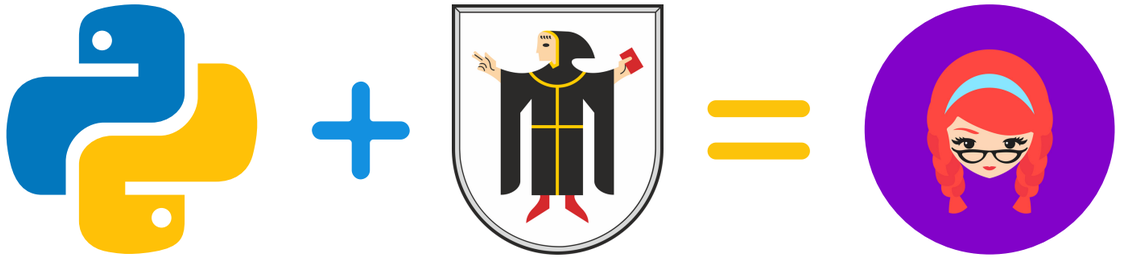
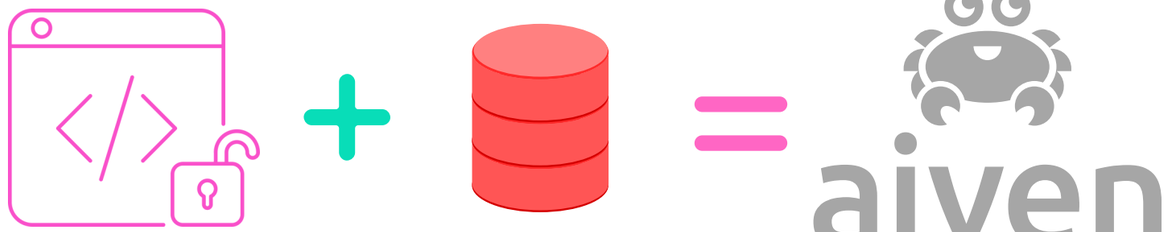
# Python 3.10: Welcome to pattern matching 🔥

@laysauchoa

# self.intro()


@laysauchoa





@aiven_io

# Structure

**Part 1:** Python 3.10

**Part 2:** Pattern Matching 

**Part 3:** Final thoughts

@laysauchoa

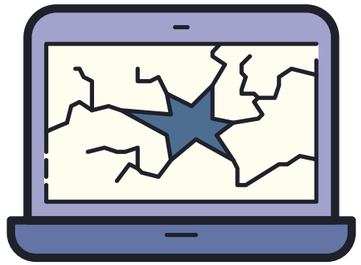
# Python 3.10

## Party


@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

# Python 3.9.7

@laysauchoa

# Python 3.10

## Meaningful messages

**Anthony Shaw**
@anthonypjshaw

Loving Python 3.10 so far. All these "little" improvements like suggesting which attribute you meant are going to make life much easier

```
try:
    1/0
except ZeroDivisionError, KeyError:
    print("Something is wrong")

  File "<ipython>", line 3
    except ZeroDivisionError, KeyError:
           ^
SyntaxError: multiple exception types must be parenthesized
```

@laysauchoa

**laysa ッ**
@laysauchoa

54.7% of the 360 most downloaded packages on PyPI supports Python 3.10 now!

: pyreadiness.org/3.10/

#Python

@laysauchoa

# Python 3.10

## mypy


@laysauchoa

**PEP 634 -- Structural Pattern Matching: Specification**
**PEP 622 -- Structural Pattern Matching**
**PEP 636 -- Structural Pattern Matching: Tutorial**

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

@laysauchoa

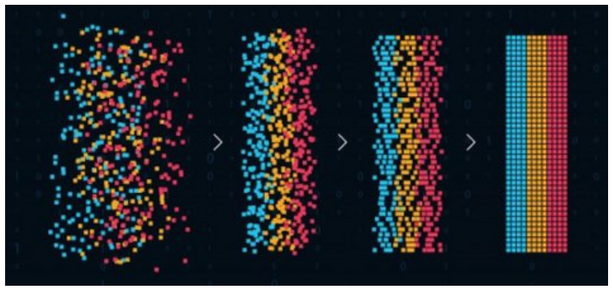
# Python 3.10

## What is new?


@laysauchoa

# Pattern matching

## "either it will or will not be a match"

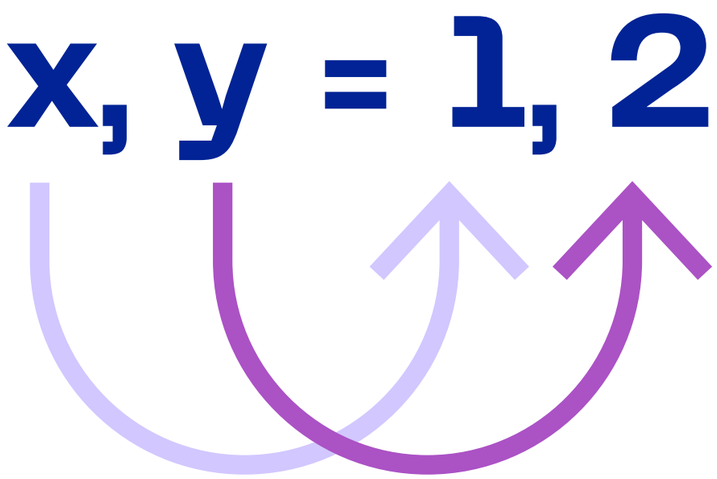



not a dating app
not a pattern recognition

@laysauchoa

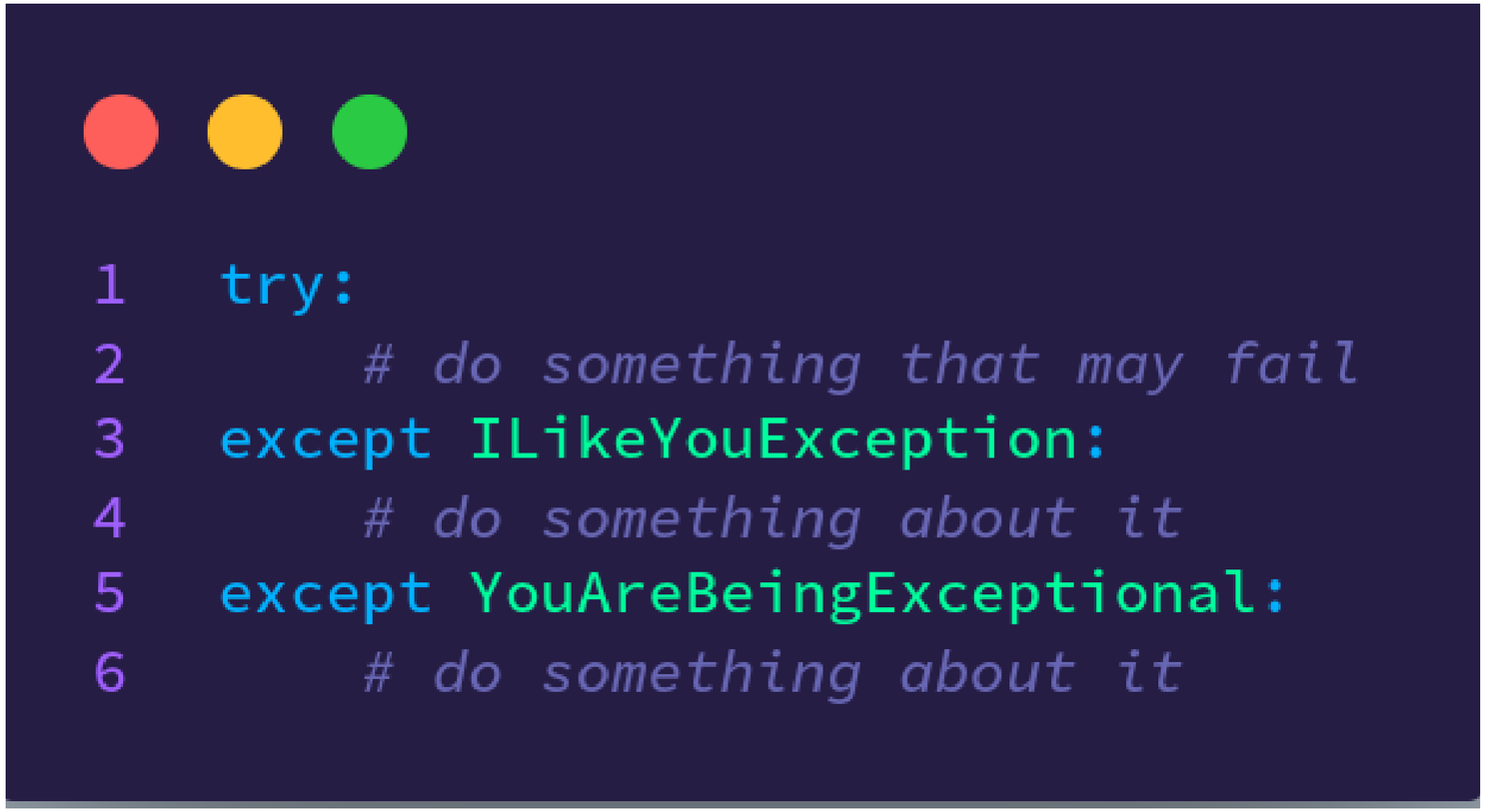
# Primitive

## Pattern matching



# Primitive

## Pattern matching



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

# Game

## Pattern matching


source: @TodoDiaGatinhos

@laysauchoa

# Steckspielzeug

@laysauchoa

# First Game

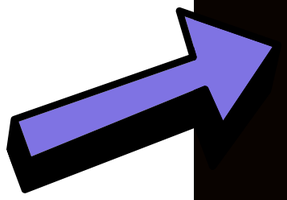
## Pattern matching

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

@laysauchoa

# Pattern matching

pattern



```
form = "triangle"
match form:
    case "circle":
        print("Circle: ")
    case "square":
        print("Square: ")
    case "triangle":
        print("Triangle: ")
    case _:
        print("Any other form")
```

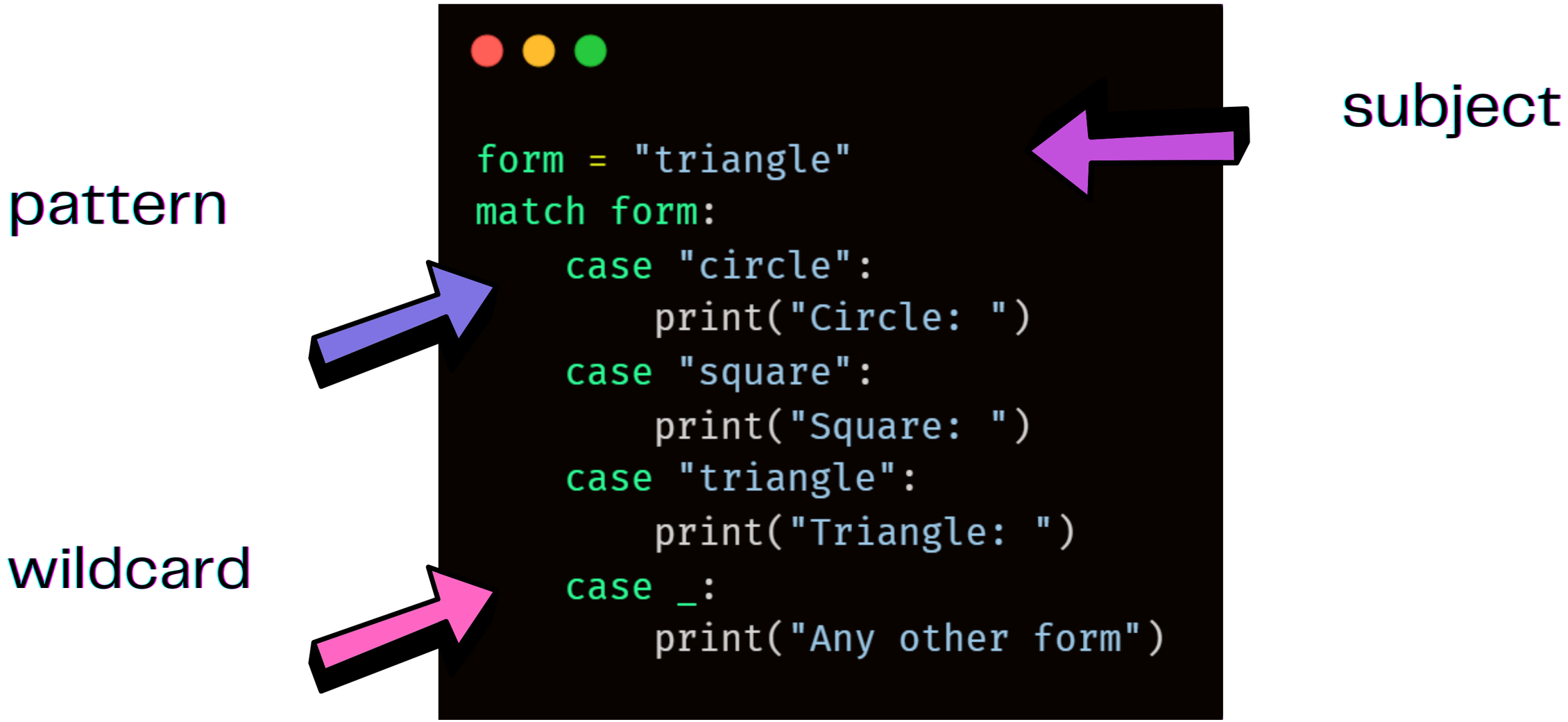
# Pattern matching

pattern

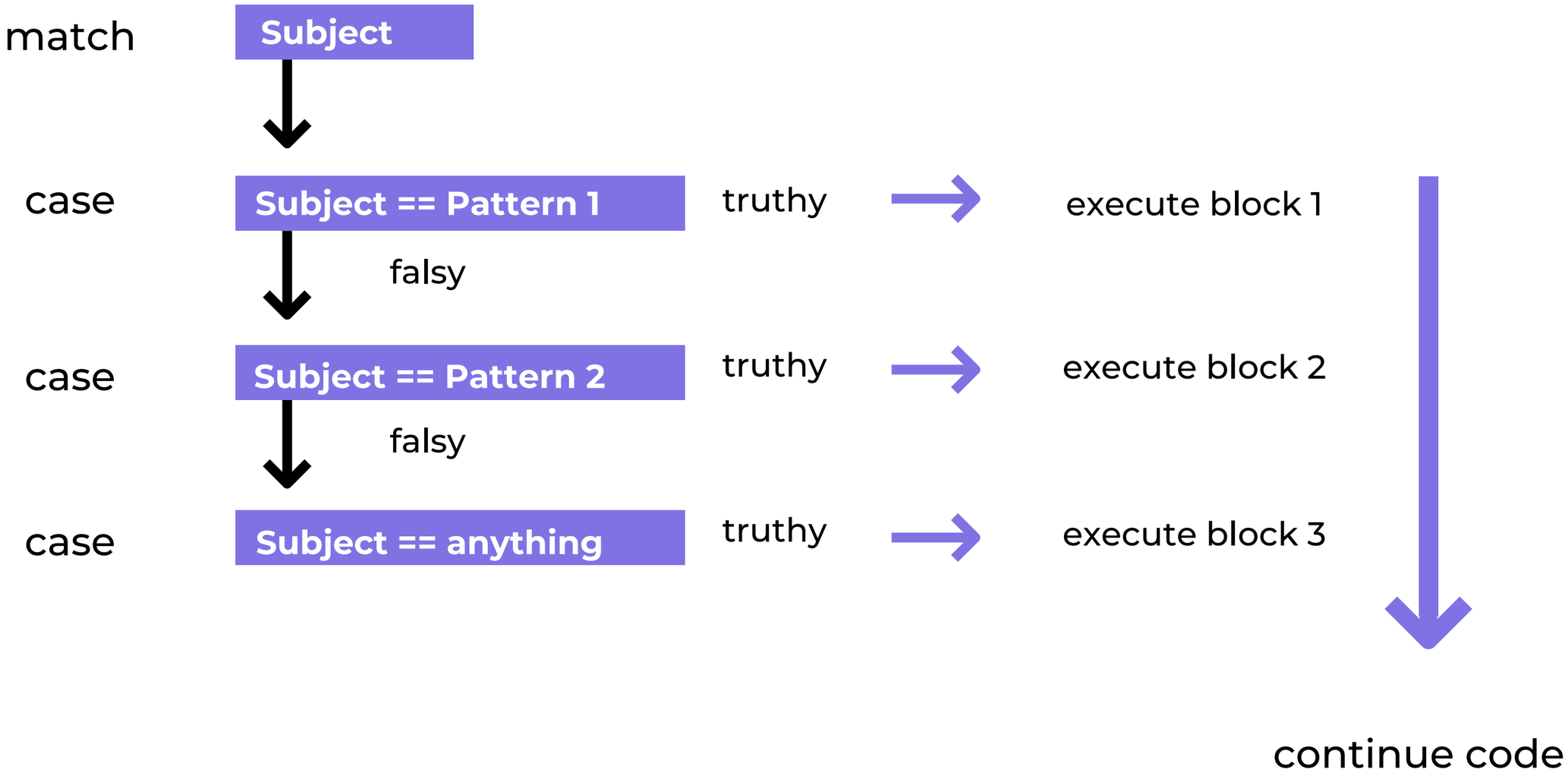
```
form = "triangle"
match form:
    case "circle":
        print("Circle: ")
    case "square":
        print("Square: ")
    case "triangle":
        print("Triangle: ")
    case _:
        print("Any other form")
```

subject

# Pattern matching



# Pattern matching



# Don'ts X

## Pattern matching

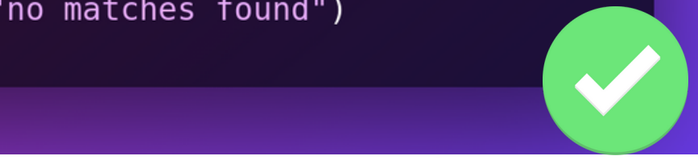
```
In [1]: form = "triangle"
   ...:
   ...: match form:
   ...:     case _:
   ...:         print("Any other form!")
   ...:     case "circle":
   ...:         print("Circle: ⚪")
   ...:     case "square":
   ...:         print("Square: ⬜")
   ...:     case "triangle":
   ...:         print("Triangle: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

@laysauchoa

# With and Without

## Pattern matching

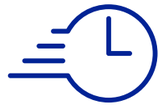
```python
def store_elements(elements: List[str])-> str:
    match elements:
        case ["triangle"]:
            print("triangle is stored")
        case ["triangle", obj]:
            print(f"triangle and {obj} are stored")
        case _:
            print("no matches found")
```



```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")
    else:
        print("no matches found")
```

@laysauchoa

# %timeit



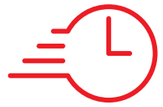
## Pattern matching

```
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)
```

@laysauchoa

# %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)
```

# With and Without

## Pattern matching

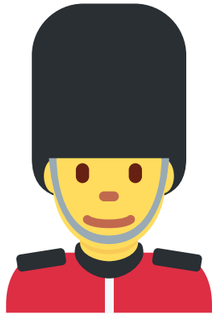
```python
def count_elements(number: int) -> str:
    match number:
        case 0:
            print("No elements!")
        case 1:
            print("One element!")
        case 2:
            print("Two elements!")
        case _:
            print("wildcard!")
```

```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!")
```

@laysauchoa

# Guard

## Pattern matching



```python
from typing import List
def match_guard(form: List):
    exists = ["triangle", "circle", "heart", "square", "rectangle"]
    match form:
        case ["form", element] if element in exists:
            print(f"Element is of form {element}.")
        case _:
            print("No matches found.")
```

@laysauchoa

# Composable Patterns with |

## Pattern matching

top

# Composable Patterns with |

## Pattern matching

top

right

# Composable Patterns with |

## Pattern matching

top

right

bottom

# Composable Patterns with |

## Pattern matching

top

left

right

bottom

# Composable Patterns with |

## Pattern matching

top

left

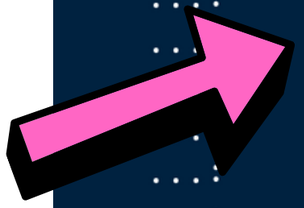
right

bottom

```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



```
In [1]:
   ...: @dataclass
   ...: class Circle:
   ...:     color: str
   ...:     radius: float
   ...:
   ...: element = Circle("pink", 10.0)
   ...:
   ...: match element:
   ...:     case Circle(color="red"):
   ...:         print(f"Red circle found.")
   ...:     case Circle(color="pink"):
   ...:         print(f"Pink circle found.")
   ...:     case Circle(color=some_color, radius=value):
   ...:         print(f"Circle found with color: {some_color} and radius: {value}")
   ...:
Pink circle found.
```

@laysauchoa

# Python 3.10

## Dict Pattern matching

```
pikachu_data = {
    "name": "Pikachu",
    "type": "Electric",
    "evolve": "Raichu"
  }
match pikachu_data
  case {"type":"Psychic"}
    print("🔎 🧠")
  case {"type":"Water"}
    print("🌊 🌊 🌊 ")
  case {"type":"Electric"}
    print("⚡⚡⚡⚡")
  case _
    print("Unknown type")
```

@laysauchoa

# Problem

## Match all "matchme-"

```
data = [{
          "id"      : "matchme-foo",
          "message": "hallo this is a message",
      },{
          "id"      : "matchme-bar",
          "message": "goodbye",
      },{
          "id"      : "anotherid",
          "message": "completely diffrent event"
      }, ...]
```

https://stackoverflow.com/questions/70111401/python-3-10-pattern-matching-pep-634-wildcard-in-string

@laysauchoa

# Problem

## Match all "matchme-"

```
for event in data:
    match event:
        case {'id': x} if x.startswith("matchme"):
            print(event["message"])
        case {'id':'anotherid'}:
            print(event["message"])
```

https://stackoverflow.com/questions/70111401/python-3-10-pattern-matching-pep-634-wildcard-in-string

@laysauchoa

# 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


@laysauchoa

# Questions

If you want to know anything more **Python 3.10** or share your ideas, **feel free to talk to me @laysauchoa or @aiven_io**

**Thank you!**