What is procedural programming?

PROGRAMMING PARADIGM CONCEPTS

Eleanor Thomas
Senior Data Analytics Engineer

What is procedural programming?

- **Procedural programming:** an imperative programming paradigm in which programs are built using *procedures*
- **Procedure (subroutine):** a series of steps that can be referenced and rerun multiple times
- **NOTE:** Procedural programming is a type of imperative programming; not all imperative programming is procedural

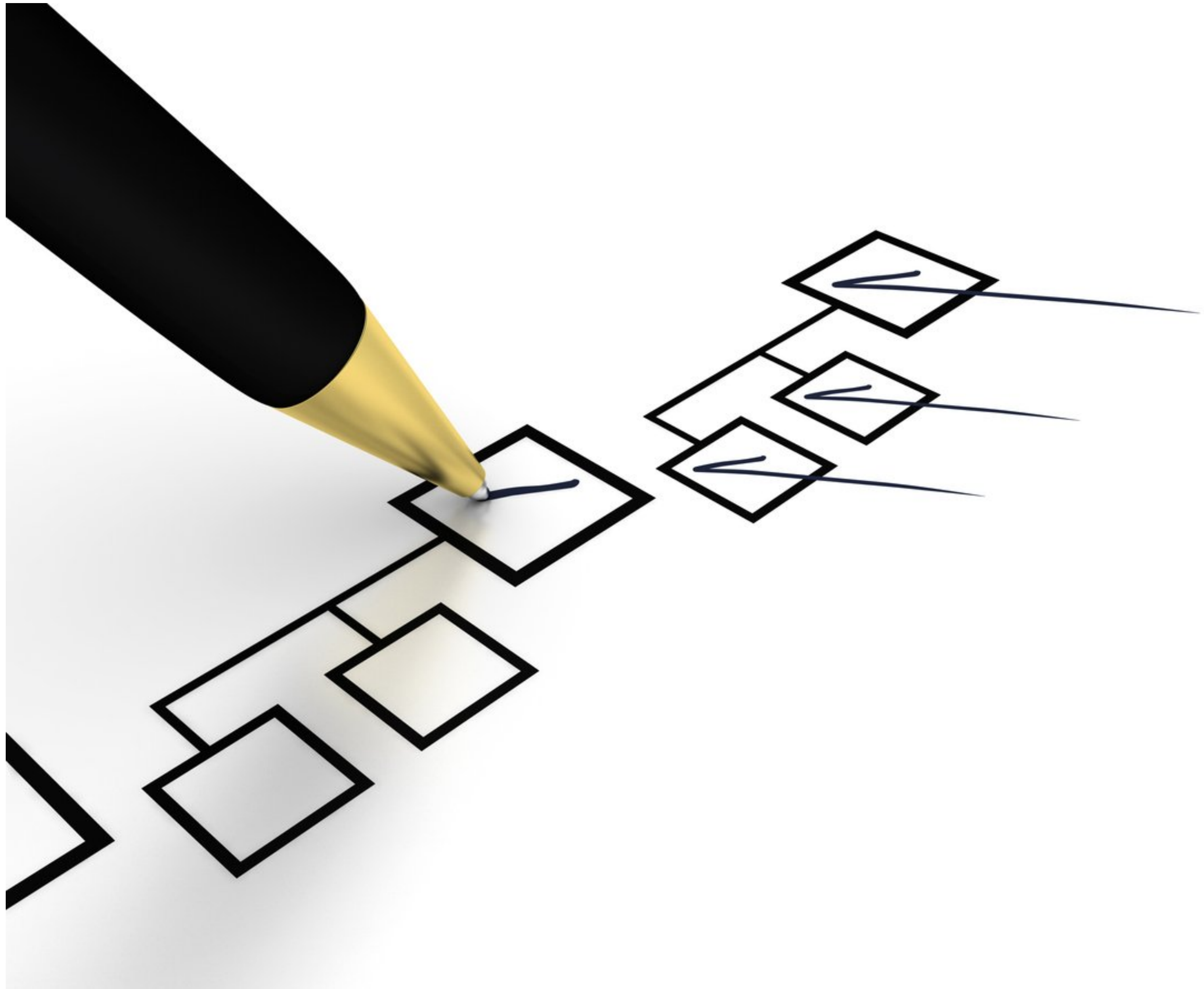
Imperative Programming

?

Procedural Programming

Procedures in procedural programming

- **Procedures:** how procedural programming achieves separation of responsibilities
- Procedures aid with modularity
- Allow for a section of code to be rerun in multiple places efficiently
- Help with organization and readability



Example of procedures

- In Python, procedures are implemented as functions

```
ducks = ['Huey', 'Dewey', 'Louie']
sorted_ducks = sort_ducks(ducks)
```

Example of a procedural program

```python
def print_initial(name):
    initial = name[0]
    print(initial)

print_initial("Marwa")
print_initial("Celia")
print_initial("Raqael")
```

Output:

```
"M"
"C"
"R"
```

Let's practice!

PROGRAMMING PARADIGM CONCEPTS

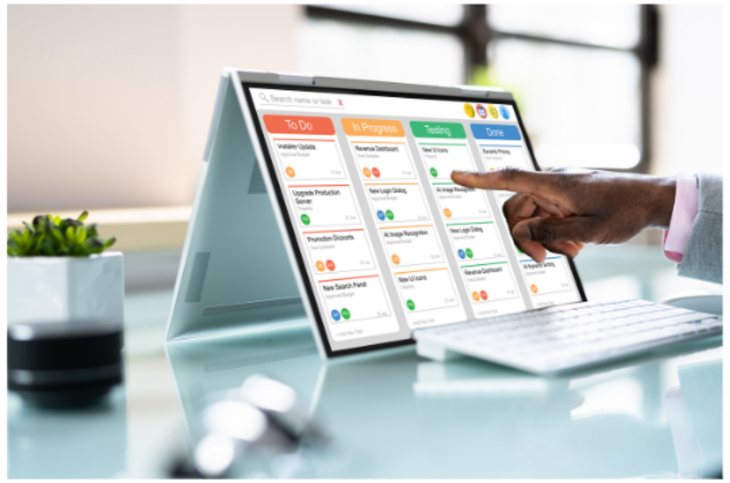
When is procedural programming used?

PROGRAMMING PARADIGM CONCEPTS

Eleanor Thomas
Senior Data Analytics Engineer

Typical applications for procedural programming

- Commonly used in general-purpose programming languages
- Makes sense when problems naturally break down into steps
- Great for automating data preparation processes
- Used in building websites
- Any kind of coding task where the problem can be defined step-by-step and broken up into subtasks



Procedural programming example

```
price_in_cents = 500
price_in_dollars = price_in_cents / 100

discount = 0.20
final_price = price_in_dollars * (1 - discount)
print("The final price is: $", final_price)
```

- Start with the price in cents
- Convert to dollars
- Apply a discount
- Print the result

Pros and cons of procedural programming

PROS

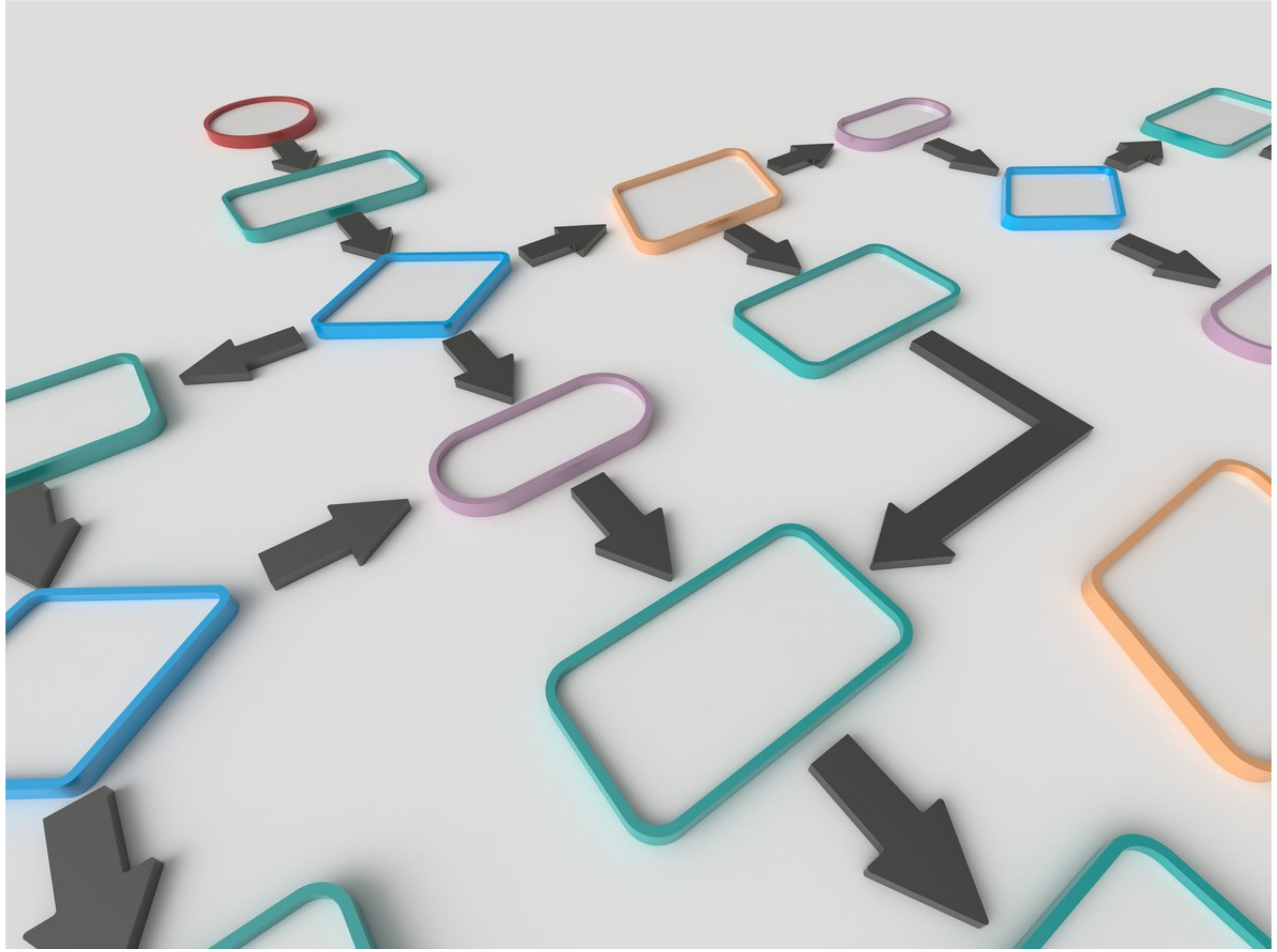
- Benefits modularity, reducing overall amount of code and saving time
- More straightforward than other paradigms, accessible, commonly used with a lot of educational resources available
- Highly flexible and appropriate for many use cases

CONS

- Can be less secure than other paradigms due to how data is moved around
- Code is rarely reusable *between* projects
- Focus is on the operations to be done with the data rather than the integrity of the data

Procedural programming vs. imperative programming

- Procedural programming is imperative programming, but ***not all*** imperative programming is procedural
- All imperative programming makes use of step-by-step instructions on how to execute
- Procedural programming specifically makes use of procedures (or subroutines) to organize code and dictate the flow of the program



Let's practice!

PROGRAMMING PARADIGM CONCEPTS

Control flow in procedural programming

PROGRAMMING PARADIGM CONCEPTS

Eleanor Thomas
Senior Data Analytics Engineer

What is control flow?

- **Control flow:** set of keywords and processes within a programming language that indicate how the logic of the code should be stepped through
- **Examples** of control flow statements:
  - `if` / `else` statements
  - `for` loops and `while` loops
  - Function definition using `def`

Combining elements of control flow (if statements and for loops)

```
if my_height > your_height:
    print("I'm taller!")
```

```
for height in height_list:
    print(height)
```

Combined:

```
for height in height_list:
    if my_height > height:
        print("I am taller than ", height)
```

Combining elements of control flow in functions

```python
def compare_heights(my_height, height_list):
    for height in height_list:
        if my_height > height:
            print("I am taller than ", height)
    return

my_height = 63
height_list = [62, 67, 70]

compare_heights(my_height, height_list)
```

Control flow in procedural programming

- Control flow statements make procedural programming possible in Python
- Structure program logic in a step-by-step way with reusable chunks of logic
- Code this logic in Python with `if` statements, loops, and Python functions
- Control flow creates procedures which implement separation of responsibilities

Let's practice!

PROGRAMMING PARADIGM CONCEPTS