



MVUP01: A Short Course in Python Basics

Presented by Gabriel Rodrigues Palma

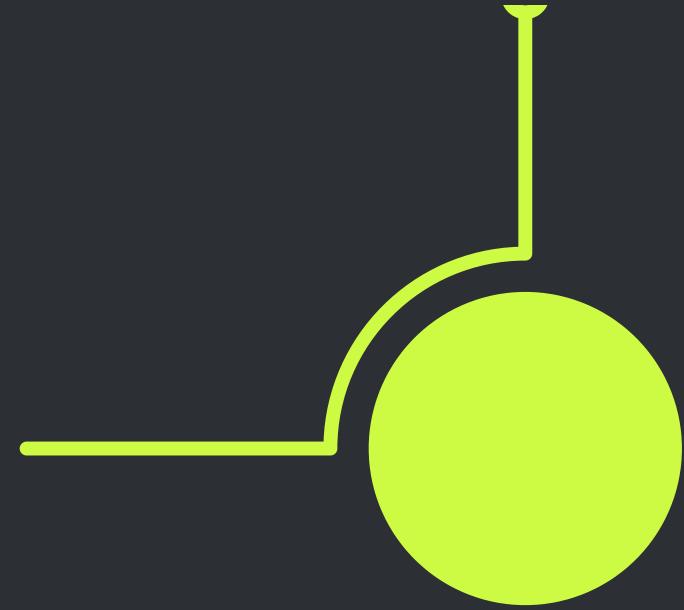


Machine Vision
using Python (MVUP01)



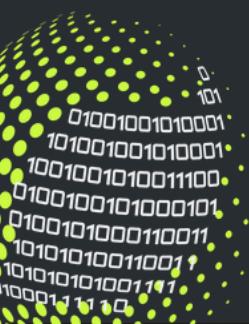
Morning Session (9:30 - 12:00)

Your First Steps into
Python (9:45 - 10:00)



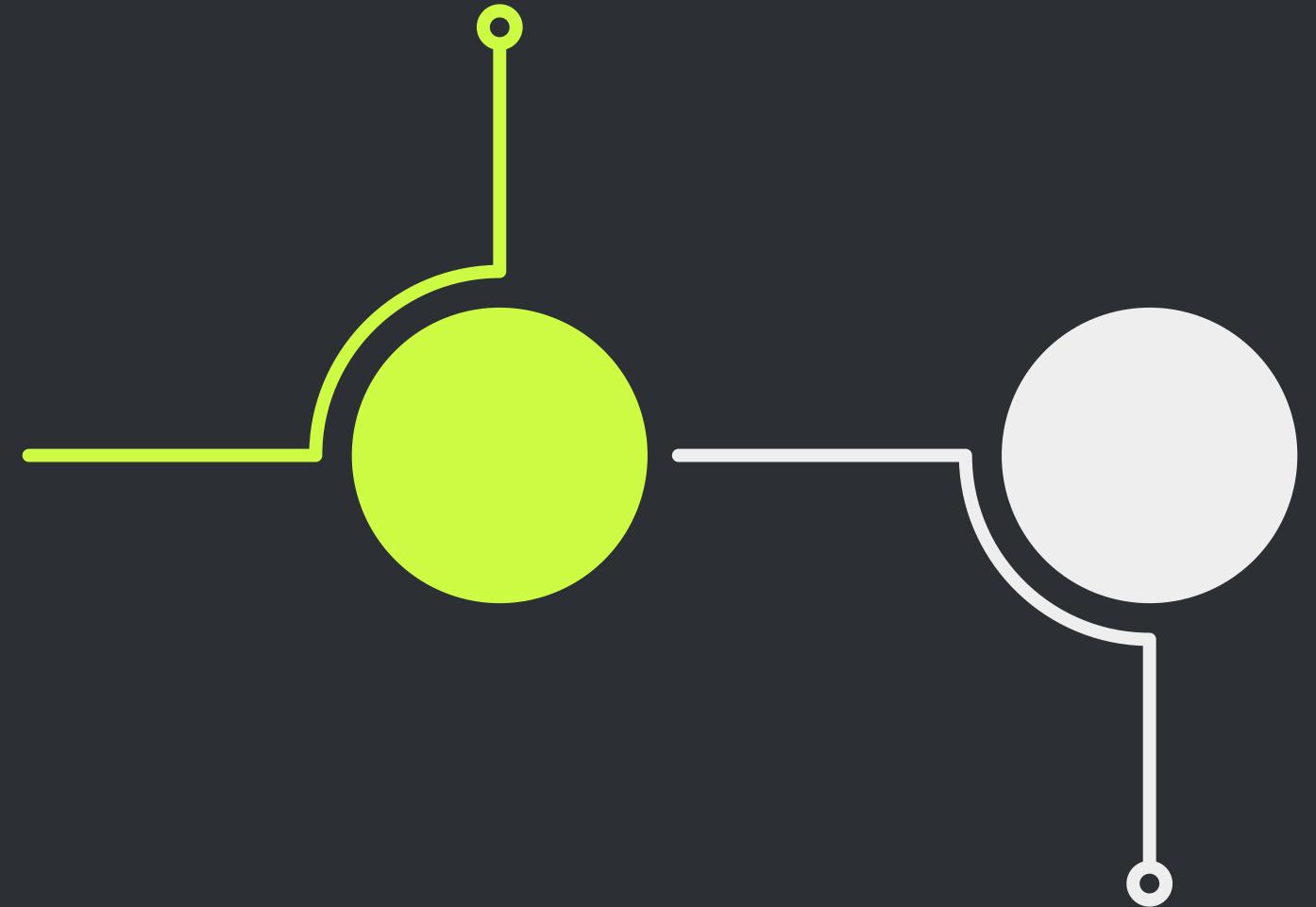
Machine Vision
using Python (MVUP01)

R stats



Morning Session (9:30 - 12:00)

Your First Steps into
Python (9:45 - 10:00)



Python programming
language (10:00 - 10:45)

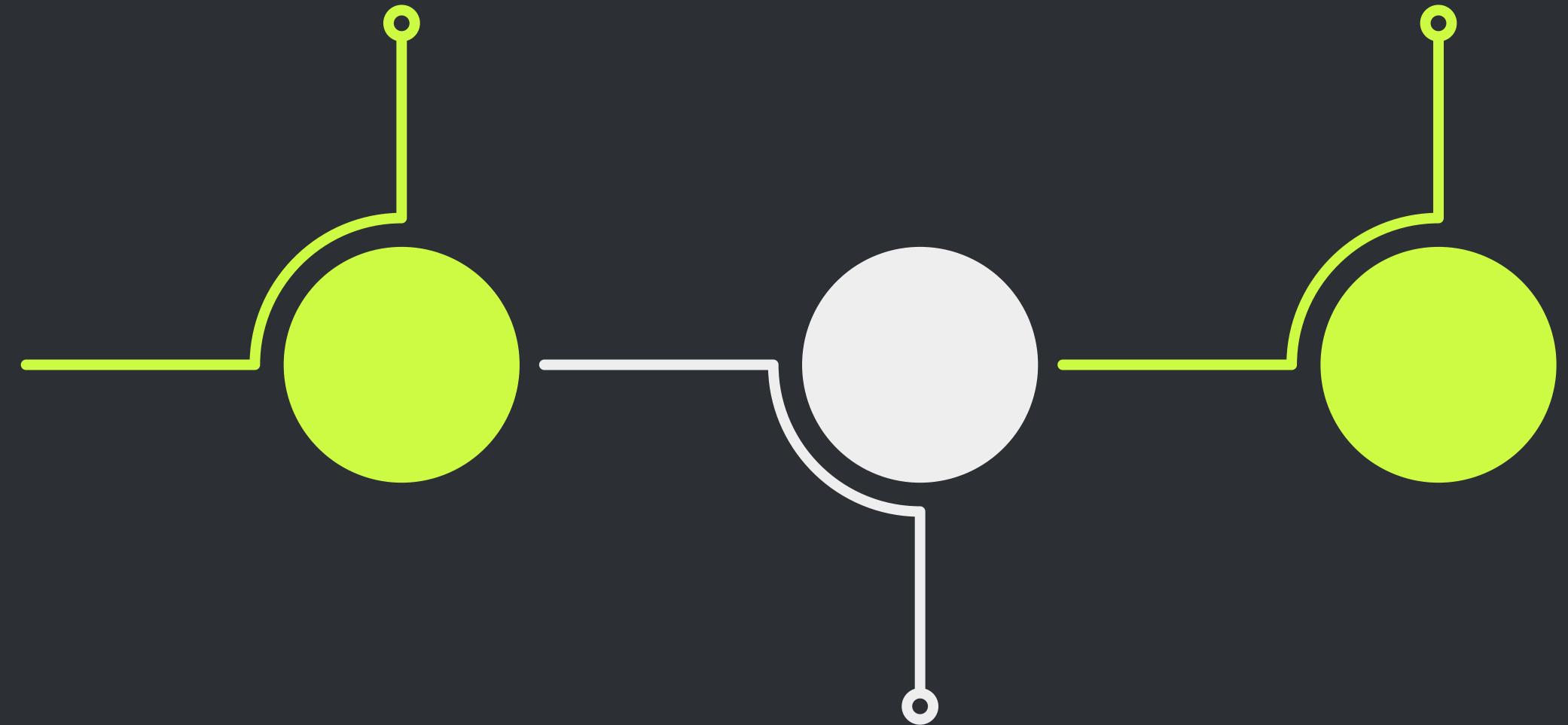
Machine Vision
using Python (MVUP01)



Morning Session (9:30 - 12:00)

Your First Steps into
Python (9:45 - 10:00)

Making Decisions in
Code (10:45 - 11:15)



Python programming
language (10:00 - 10:45)

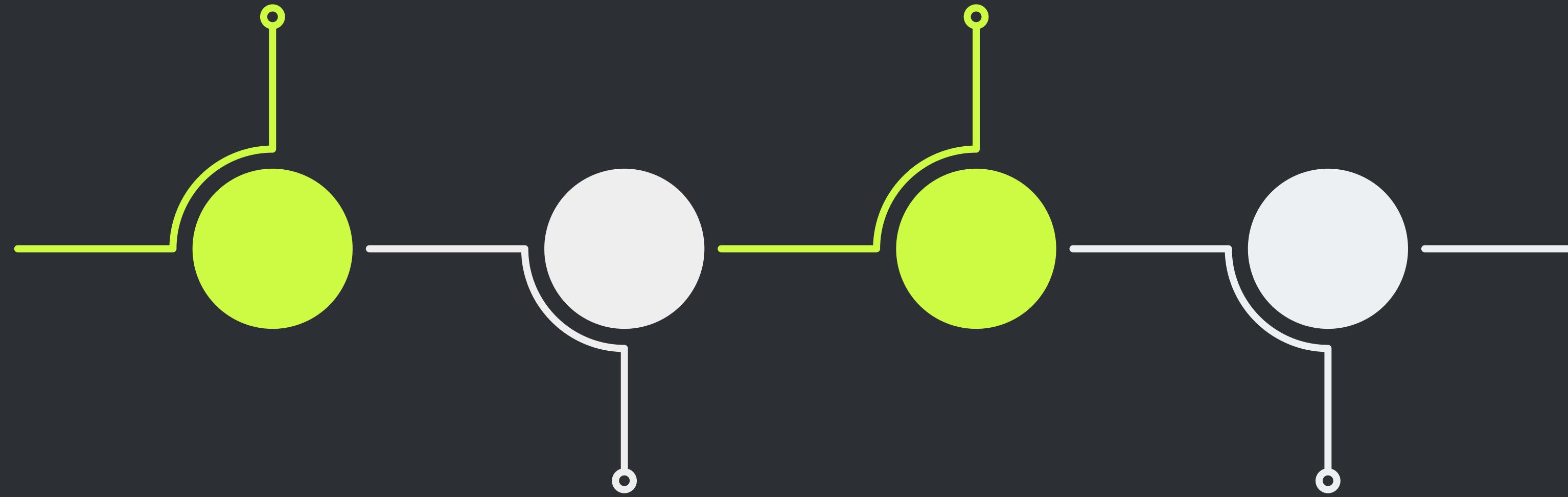
Machine Vision
using Python (MVUP01)



Morning Session (9:30 - 12:00)

Your First Steps into
Python (9:45 - 10:00)

Making Decisions in
Code (10:45 - 11:15)



Python programming
language (10:00 - 10:45)

The Power of Repetition
(11:15 - 12:00)

Machine Vision
using Python (MVUP01)





Your First Steps into Python

Machine Vision
using Python (MVUP01)



Your First Steps into Python

Machine Vision
using Python (MVUP01)



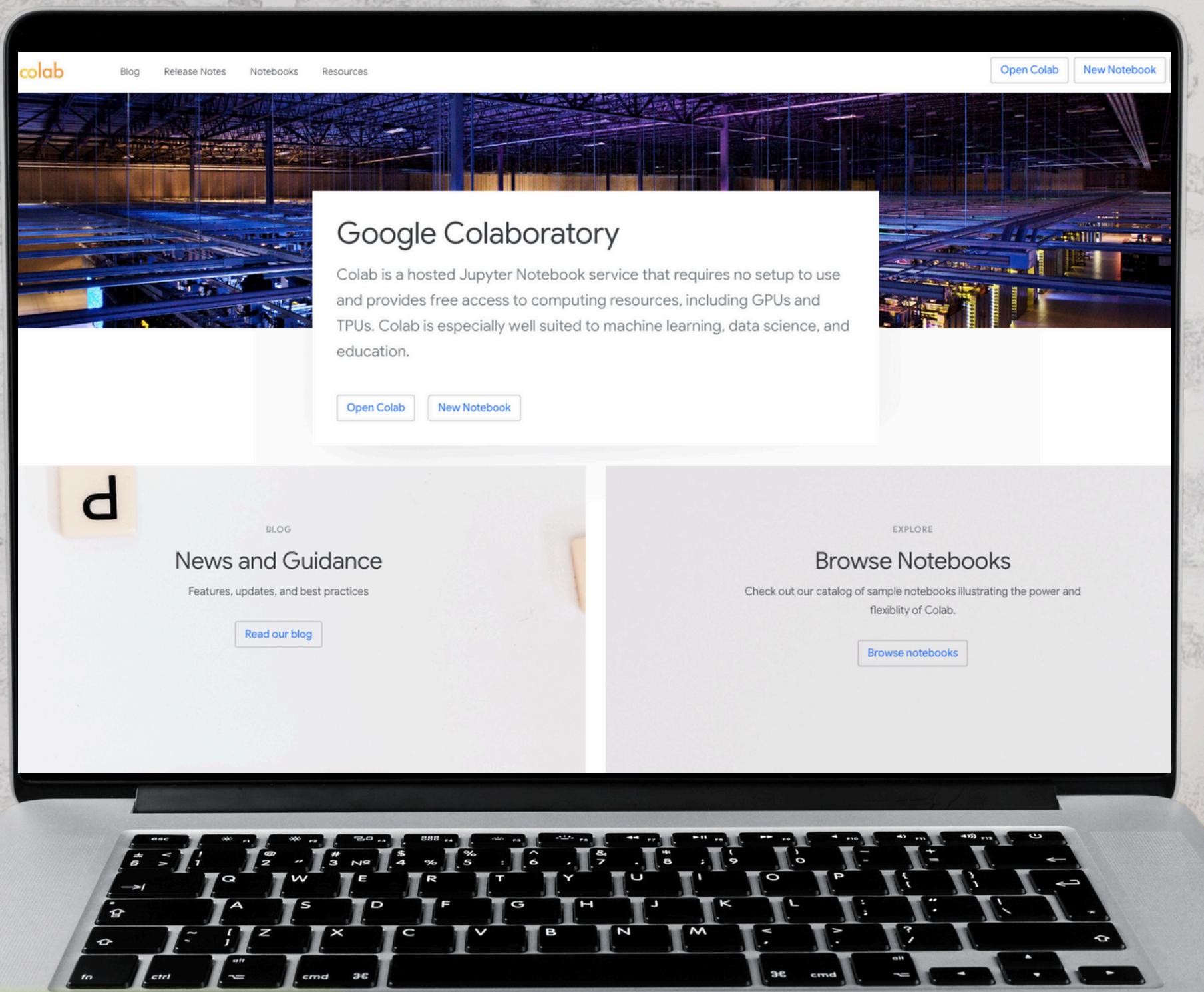
Your First Steps into Python

```
// Types can be a map of types/handlers
if ( typeof types === "object" ) {
    // types-Object, selector, data
    if ( typeof selector !== "string" ) {
        data = data || selector;
        selector = undefined;
    }
    for ( type in types ) {
        on( elem, type, selector, data, types[ type ], one );
    }
    return elem;
}
if ( data == null && fn == null ) {
```

Machine Vision
using Python (MVUP01)



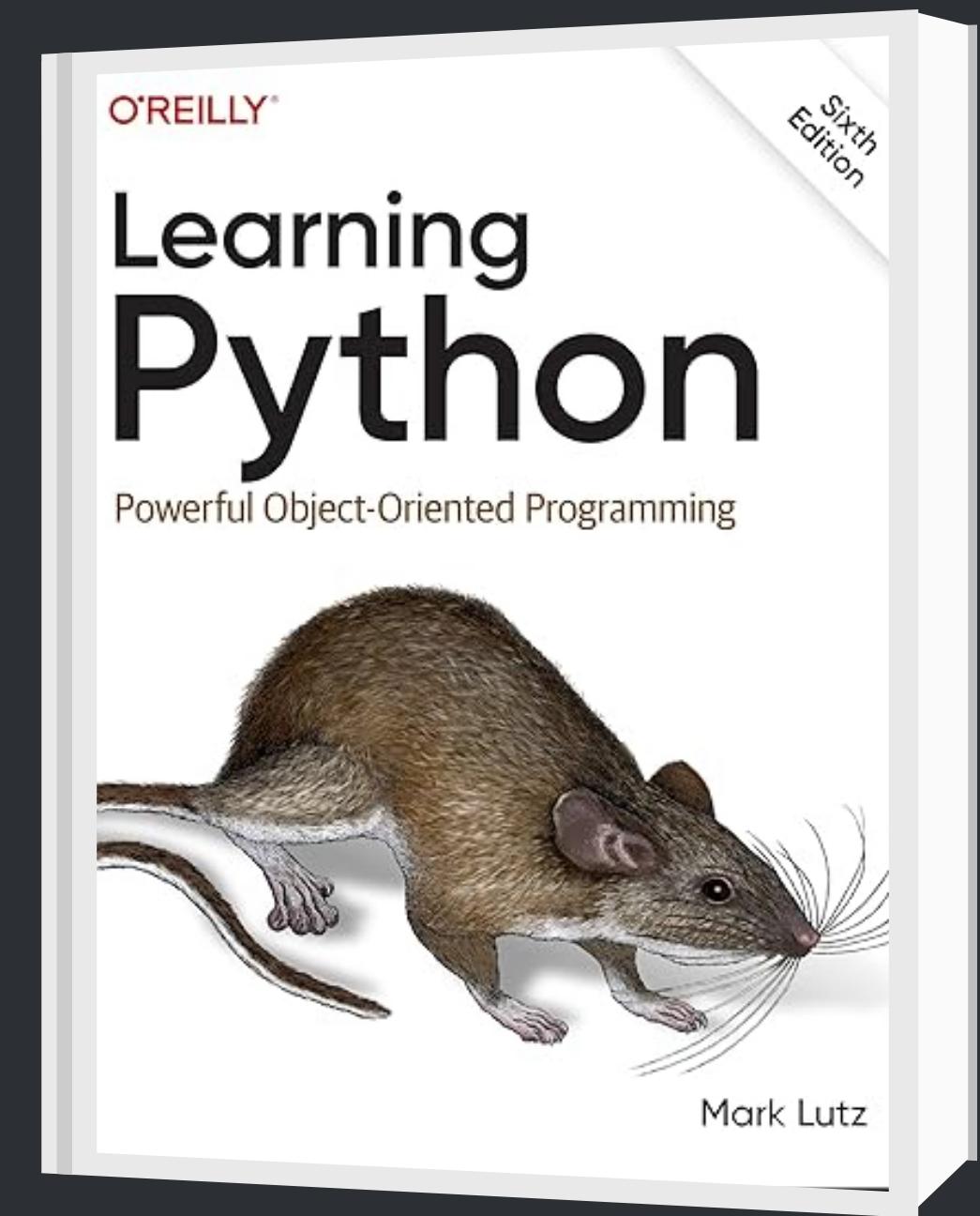
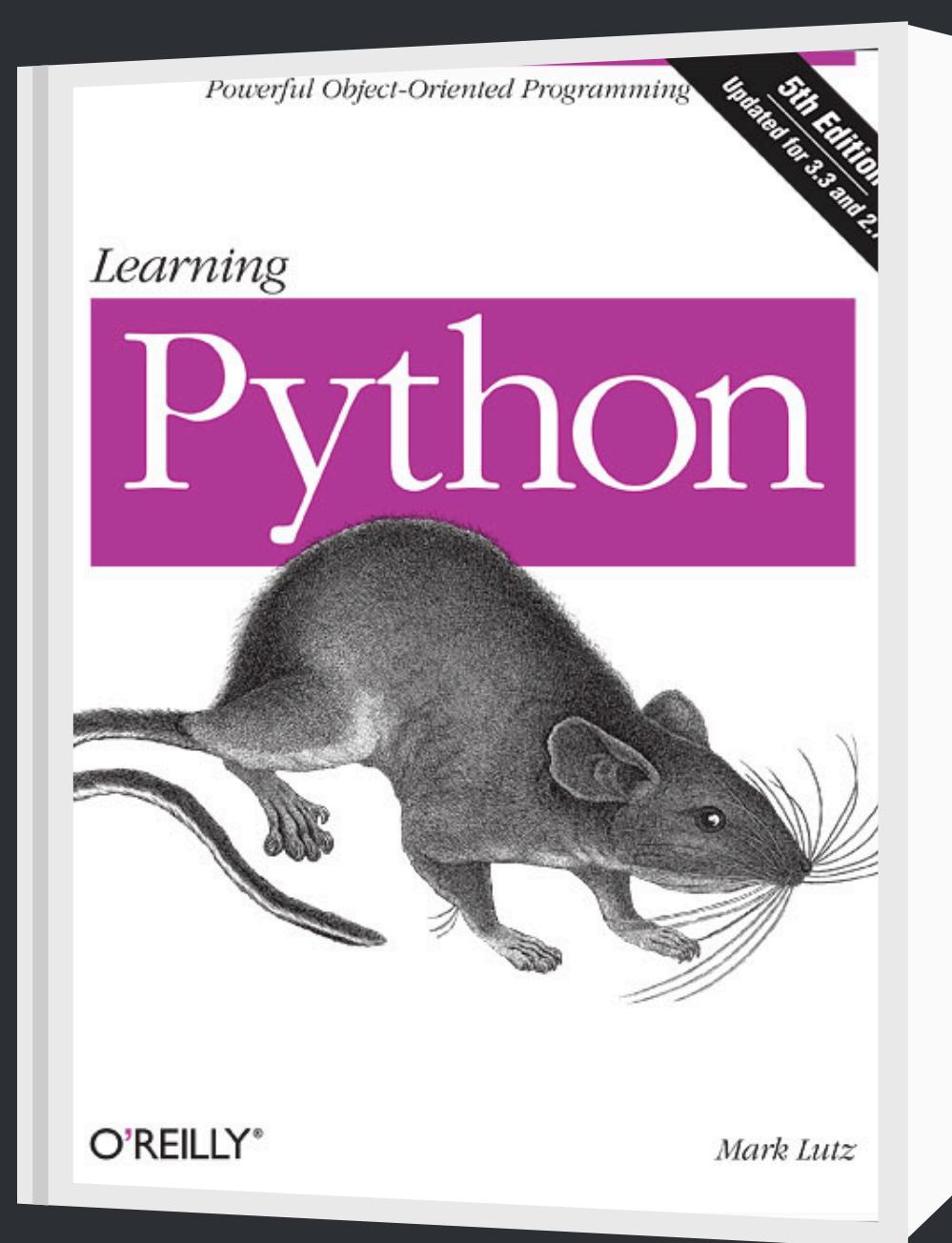
Your First Steps into Python



Machine Vision
using Python (MVUP01)



Your First Steps into Python



Machine Vision
using Python (MVUP01)

R stats



Your First Steps into Python

Code presentation:

```
# First message with Python  
print('Machine Vision With Python')
```

Input Code

Machine Vision
using Python (MVUP01)



Your First Steps into Python

Code presentation:

```
# First message with Python  
print('Machine Vision With Python')
```

Input

Machine Vision With Python

Output

Machine Vision
using Python (MVUP01)



Your First Steps into Python

Code presentation:

```
# First message with Python  
print('Machine Vision With Python')
```

Input

Machine Vision With Python

Output

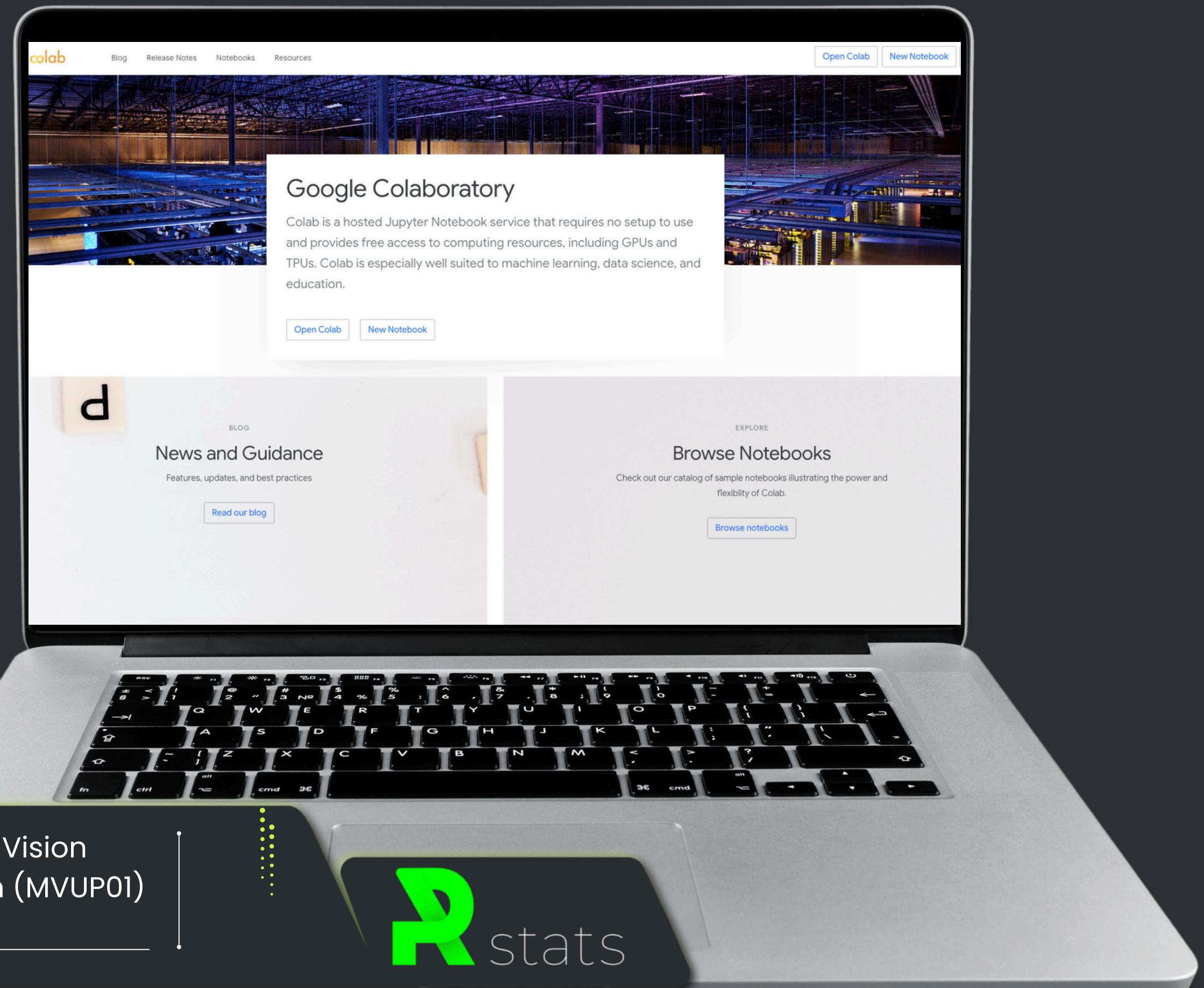
We are going to be using this structure for the rest of the course.

Machine Vision
using Python (MVUP01)



Your First Steps into Python

Hands-on activity:



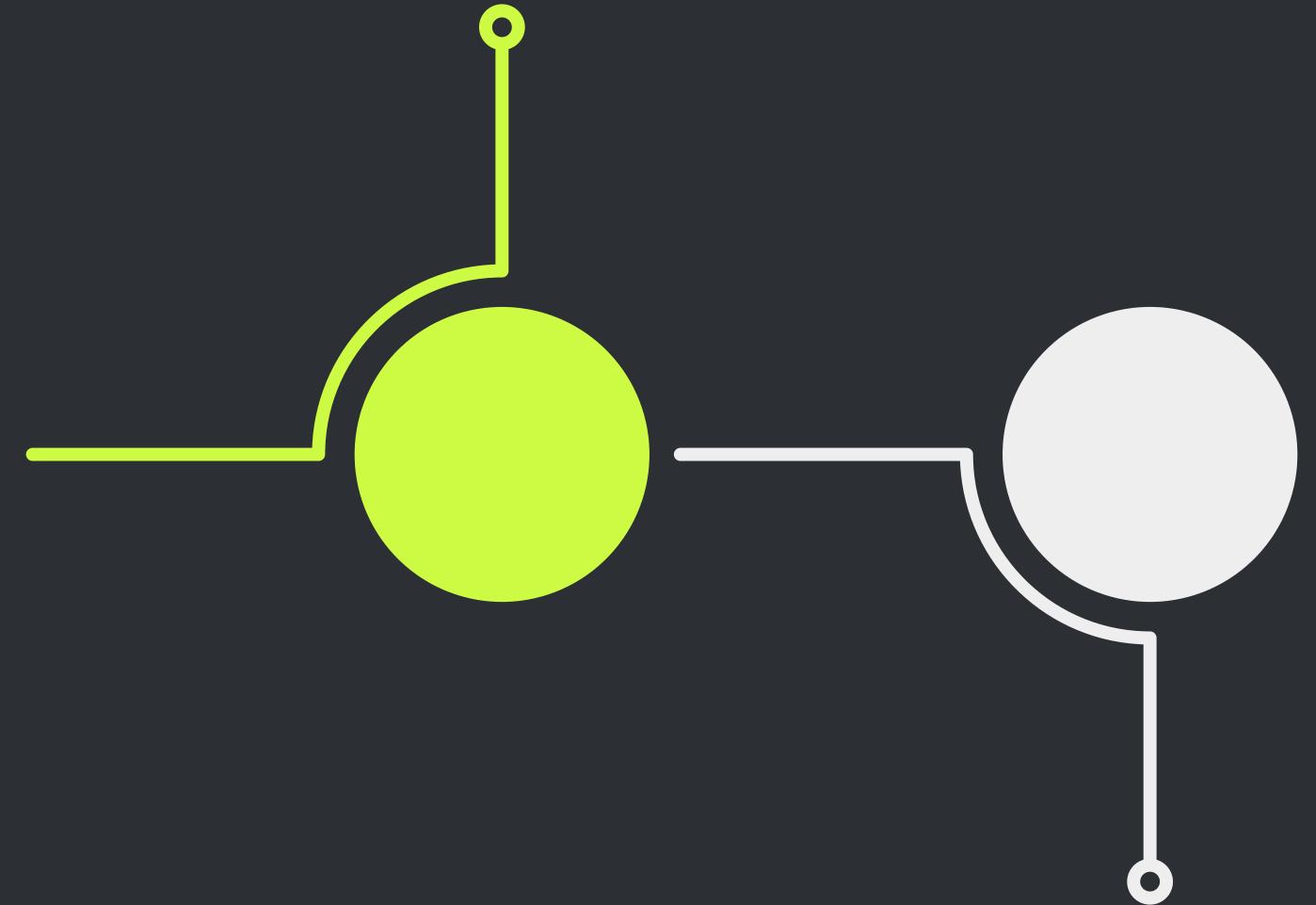
Machine Vision
using Python (MVUP01)

R stats



Morning Session (9:30 - 12:00)

Your First Steps into
Python (9:45 - 10:00)



Python programming
language (10:00 - 10:45)

Machine Vision
using Python (MVUP01)



Python programming language



Machine Vision
using Python (MVUP01)



Python programming language



Machine Vision
using Python (MVUP01)



Python programming language

Python syntax fundamentals (Variables)

```
# Example 1: Variables and Basic Data Types  
species_name = "Panthera onca"      # String (text)
```

Python programming language

Python syntax fundamentals (Variables)

```
# Example 1: Variables and Basic Data Types  
species_name = "Panthera onca"      # String (text)
```

species_name : Name of the variable (case-sensitive);

Python programming language

Python syntax fundamentals (Variables)

```
# Example 1: Variables and Basic Data Types  
species_name = "Panthera onca"      # String (text)
```

`species_name` : Name of the variable (case-sensitive);

`=` : For assigning a value;

Python programming language

Python syntax fundamentals (Variables)

```
# Example 1: Variables and Basic Data Types  
species_name = "Panthera onca"      # String (text)
```

`species_name` : Name of the variable (case-sensitive);

`=` : For assigning a value;

`"Panthera onca"` : The value to assigned to the variable.

Python programming language

Python syntax fundamentals (Variables)

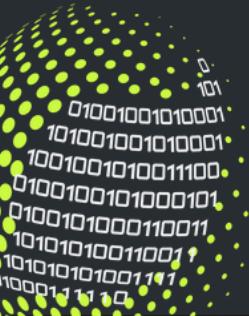
```
# Example 1: Variables and Basic Data Types
species_name = "Panthera onca"      # String (text)
print(f"Species: {species_name} | Type: {type(species_name)}")
```

Python programming language

Python syntax fundamentals (Variables)

```
# Example 1: Variables and Basic Data Types
species_name = "Panthera onca"          # String (text)
print(f"Species: {species_name} | Type: {type(species_name)}")
```

```
Species: Panthera onca | Type: <class 'str'>
```



Machine Vision
using Python (MVUP01)



Python programming language

Python syntax fundamentals (Variables)

```
population_size = 200                      # Integer (whole number)
print(f"Population: {population_size} | Type: {type(population_size)}")
```

Python programming language

Python syntax fundamentals (Variables)

```
population_size = 200          # Integer (whole number)
print(f"Population: {population_size} | Type: {type(population_size)}")
```

```
Population: 200 | Type: <class 'int'>
```

Python programming language

Python syntax fundamentals (Variables)

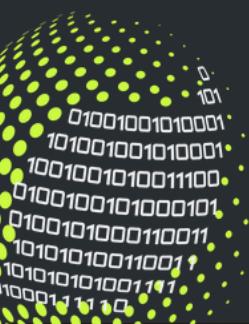
```
survival_rate = 0.85          # Float (decimal)
print(f"Survival Rate: {survival_rate} | Type: {type(survival_rate)}")
```

Python programming language

Python syntax fundamentals (Variables)

```
survival_rate = 0.85          # Float (decimal)
print(f"Survival Rate: {survival_rate} | Type: {type(survival_rate)}")
```

```
Survival Rate: 0.85 | Type: <class 'float'>
```



Python programming language

Python syntax fundamentals (Variables)

```
is_endangered = True          # Boolean (True/False)
print(f"Endangered Status: {is_endangered} | Type: {type(is_endangered)}")
```

Machine Vision
using Python (MVUP01)



Python programming language

Python syntax fundamentals (Variables)

```
is_endangered = True          # Boolean (True/False)  
print(f"Endangered Status: {is_endangered} | Type: {type(is_endangered)}")
```

```
Endangered Status: True | Type: <class 'bool'>
```

Python programming language

Python syntax fundamentals (Basic operations)

```
temperature_c = 25.5
temperature_f = temperature_c * 1.8 + 32    # Converting to Fahrenheit

print(f"Temperature (F): {temperature_f} | Type: {type(temperature_f)}")
```

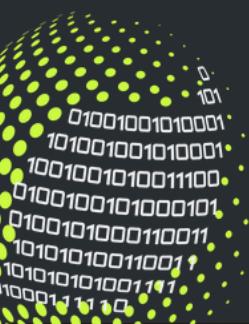
Python programming language

Python syntax fundamentals (Basic operations)

```
temperature_c = 25.5
temperature_f = temperature_c * 1.8 + 32    # Converting to Fahrenheit

print(f"Temperature (F): {temperature_f} | Type: {type(temperature_f)}")
```

```
Temperature (F): 77.9 | Type: <class 'float'>
```



Machine Vision
using Python (MVUP01)



Python programming language

Python syntax fundamentals (Basic operations)

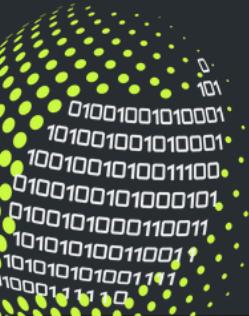
```
# Subtraction: Population decline
initial_count = 23
mortality = 4
surviving_individuals = initial_count - mortality
print(f"Surviving trees after frost: {surviving_individuals}")
```

Python programming language

Python syntax fundamentals (Basic operations)

```
# Subtraction: Population decline  
initial_count = 23  
mortality = 4  
surviving_individuals = initial_count - mortality  
print(f"Surviving trees after frost: {surviving_individuals}")
```

Surviving trees after frost: 19



Machine Vision
using Python (MVUP01)



Python programming language

Python syntax fundamentals (Basic operations)

```
# Division: Calculate average density
total_organisms = 27
sample_area = 3
density = total_organisms / sample_area
print(f"Organism density per m2: {density}")
```

Python programming language

Python syntax fundamentals (Basic operations)

```
# Division: Calculate average density  
total_organisms = 27  
sample_area = 3  
density = total_organisms / sample_area  
print(f"Organism density per m2: {density}")
```

Organism density per m²: 9.0

Python programming language

Python syntax fundamentals (Basic operations)

```
# Modulo: DNA sequence analysis
base_pairs = 7
nucleotides = 4 # A, T, C, G
current_base = base_pairs % nucleotides
print(f"Nucleotide position (0=A, 1=T, 2=C, 3=G): {current_base}")
```

Python programming language

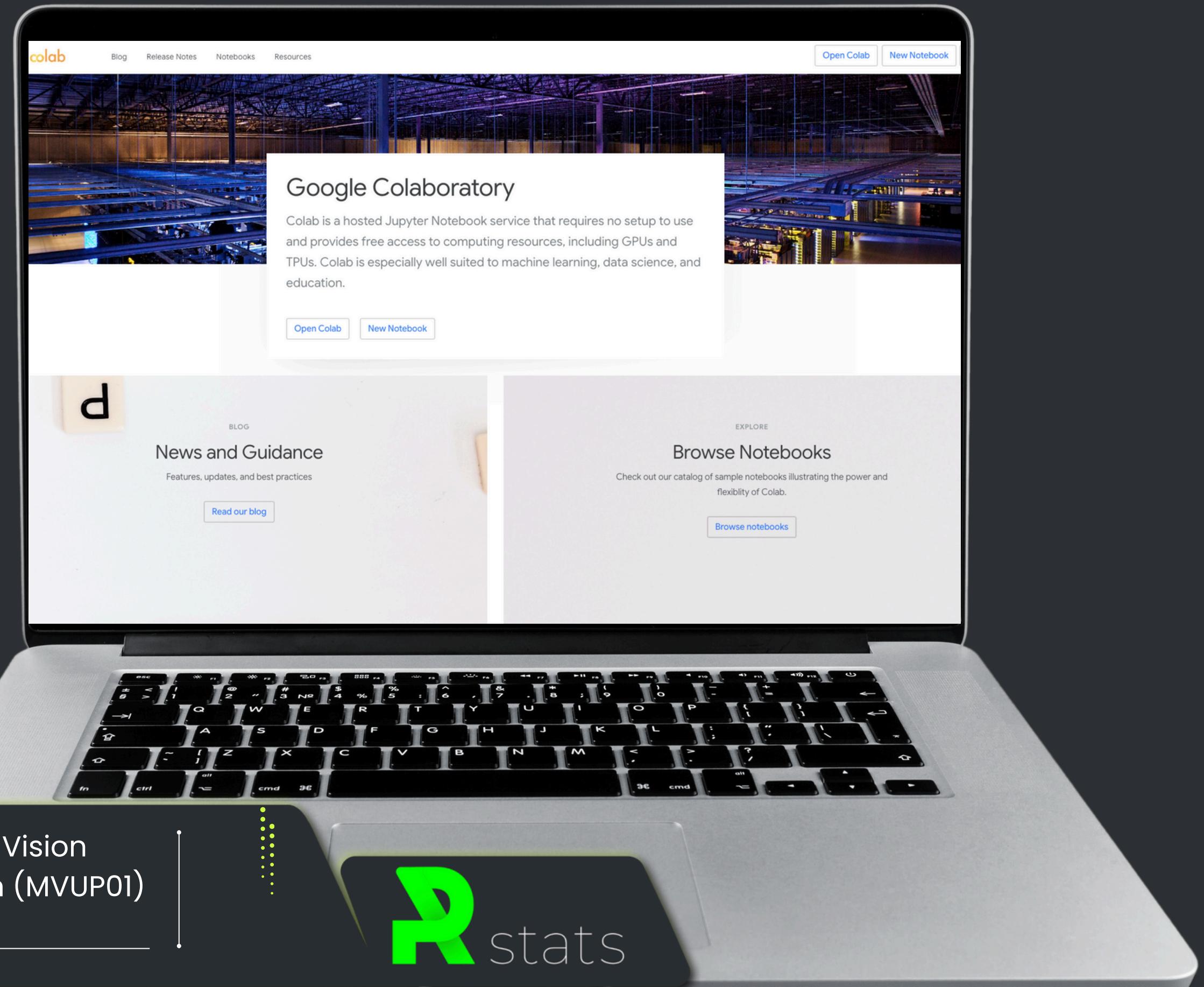
Python syntax fundamentals (Basic operations)

```
# Modulo: DNA sequence analysis  
base_pairs = 7  
nucleotides = 4 # A, T, C, G  
current_base = base_pairs % nucleotides  
print(f"Nucleotide position (0=A, 1=T, 2=C, 3=G): {current_base}")
```

Nucleotide position (0=A, 1=T, 2=C, 3=G): 3

Python programming language

Hands-on activity:



Machine Vision
using Python (MVUP01)

R stats

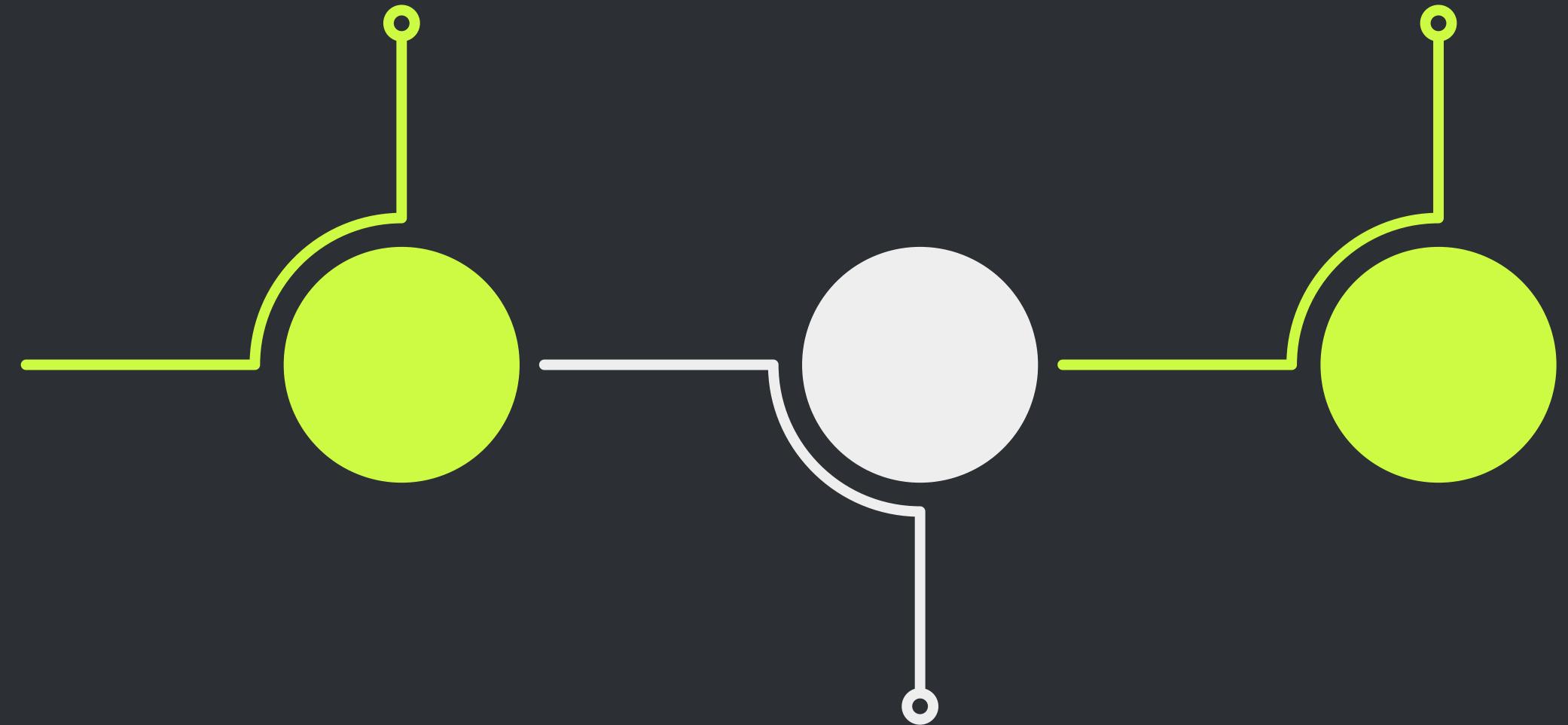
01001001010001
101001001010001
1001001010001100
0100100101000101
010010100011001
101010001100011
101010010110001
101010101001111
101010101011111



Morning Session (9:30 - 12:00)

Your First Steps into
Python (9:45 - 10:00)

Making Decisions in
Code (10:45 - 11:15)



Machine Vision
using Python (MVUP01)



Making Decisions in Code



Machine Vision
using Python (MVUP01)



Making Decisions in Code

Machine Vision
using Python (MVUP01)



Making Decisions in Code

If/Elif/else structures

```
# Input variables
wings = 2
body_segments = 3
mouthparts = "proboscis"

# Classification logic
if wings == 2 and body_segments == 3 and mouthparts == "proboscis":
    classification = "Order Diptera (Flies and Mosquitoes)"
elif wings == 4 and body_segments == 3:
    classification = "Order Lepidoptera (Butterflies and Moths)"
elif wings == 4 and mouthparts == "mandibles":
    classification = "Order Coleoptera (Beetles)"
else:
    classification = "Other Order – Further identification needed"

print(f"Insect Classification: {classification}")
```

Making Decisions in Code

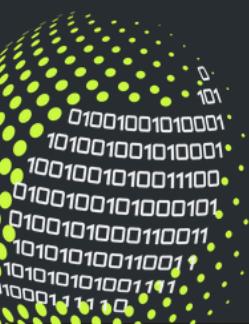
If/Elif/else structures

```
# Input variables
wings = 2
body_segments = 3
mouthparts = "proboscis"

# Classification logic
if wings == 2 and body_segments == 3 and mouthparts == "proboscis":
    classification = "Order Diptera (Flies and Mosquitoes)"
elif wings == 4 and body_segments == 3:
    classification = "Order Lepidoptera (Butterflies and Moths)"
elif wings == 4 and mouthparts == "mandibles":
    classification = "Order Coleoptera (Beetles)"
else:
    classification = "Other Order – Further identification needed"

print(f"Insect Classification: {classification}")
```

Insect Classification: Order Diptera (Flies and Mosquitoes)



Machine Vision
using Python (MVUP01)



Making Decisions in Code

If/Elif/else structures

```
# Input variables
rainfall = 800      # mm per year
elevation = 1500    # meters

if rainfall < 250:
    habitat = "Desert ecosystem"
elif rainfall < 1000 and elevation > 1000:
    habitat = "Mountain scrubland"
elif rainfall < 2000:
    habitat = "Savanna"
else:
    habitat = "Tropical rainforest"

print(f"Habitat Type: {habitat}")
```

Making Decisions in Code

If/Elif/else structures

```
# Input variables
rainfall = 800      # mm per year
elevation = 1500    # meters

if rainfall < 250:
    habitat = "Desert ecosystem"
elif rainfall < 1000 and elevation > 1000:
    habitat = "Mountain scrubland"
elif rainfall < 2000:
    habitat = "Savanna"
else:
    habitat = "Tropical rainforest"

print(f"Habitat Type: {habitat}")
```

Habitat Type: Mountain scrubland

Machine Vision
using Python (MVUP01)



Making Decisions in Code

If/Elif/else structures

```
# Input variables
population_size = 200
decline_rate = 55 # percentage

if population_size < 50 or decline_rate > 70:
    status = "Critically Endangered"
elif population_size < 250 or decline_rate > 50:
    status = "Endangered"
elif population_size < 1000 or decline_rate > 30:
    status = "Vulnerable"
else:
    status = "Least Concern"

print(f"Conservation Status: {status}")
```

Making Decisions in Code

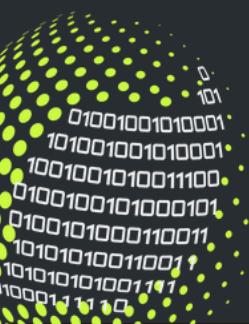
If/Elif/else structures

```
# Input variables
population_size = 200
decline_rate = 55 # percentage

if population_size < 50 or decline_rate > 70:
    status = "Critically Endangered"
elif population_size < 250 or decline_rate > 50:
    status = "Endangered"
elif population_size < 1000 or decline_rate > 30:
    status = "Vulnerable"
else:
    status = "Least Concern"

print(f"Conservation Status: {status}")
```

Conservation Status: Endangered



Machine Vision
using Python (MVUP01)



Making Decisions in Code

If/Elif/else structures

```
# Input variables
leaf_count = 2
height_cm = 15

if leaf_count == 0:
    growth_stage = "Germination stage"
elif leaf_count < 3:
    growth_stage = "Seedling stage"
elif height_cm < 30:
    growth_stage = "Vegetative stage"
elif height_cm >= 30:
    growth_stage = "Mature stage"
else:
    growth_stage = "Unknown stage"

print(f"Plant Growth Stage: {growth_stage}")
```

Making Decisions in Code

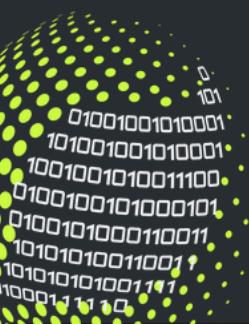
If/Elif/else structures

```
# Input variables
leaf_count = 2
height_cm = 15

if leaf_count == 0:
    growth_stage = "Germination stage"
elif leaf_count < 3:
    growth_stage = "Seedling stage"
elif height_cm < 30:
    growth_stage = "Vegetative stage"
elif height_cm >= 30:
    growth_stage = "Mature stage"
else:
    growth_stage = "Unknown stage"

print(f"Plant Growth Stage: {growth_stage}")
```

Plant Growth Stage: Seedling stage



Machine Vision
using Python (MVUP01)



Making Decisions in Code

If/Elif/else structures

```
# Input variables
elevation = 1200      # meters
rainfall = 800        # mm/year
soil_ph = 6.5
canopy_cover = 75     # percentage
human_presence = True

print("Habitat Suitability Analysis for Mountain Gorillas:")

# Combining multiple conditions with logical operators
if (elevation >= 1000 and elevation <= 3500) and \
(rainfall > 500 and rainfall < 2000):
    print("GOOD: Elevation and rainfall within suitable range")
else:
    print("WARNING: Location outside preferred elevation or rainfall range")
```

Making Decisions in Code

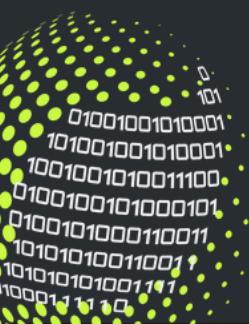
If/Elif/else structures

```
# Input variables
elevation = 1200      # meters
rainfall = 800        # mm/year
soil_ph = 6.5
canopy_cover = 75     # percentage
human_presence = True

print("Habitat Suitability Analysis for Mountain Gorillas:")

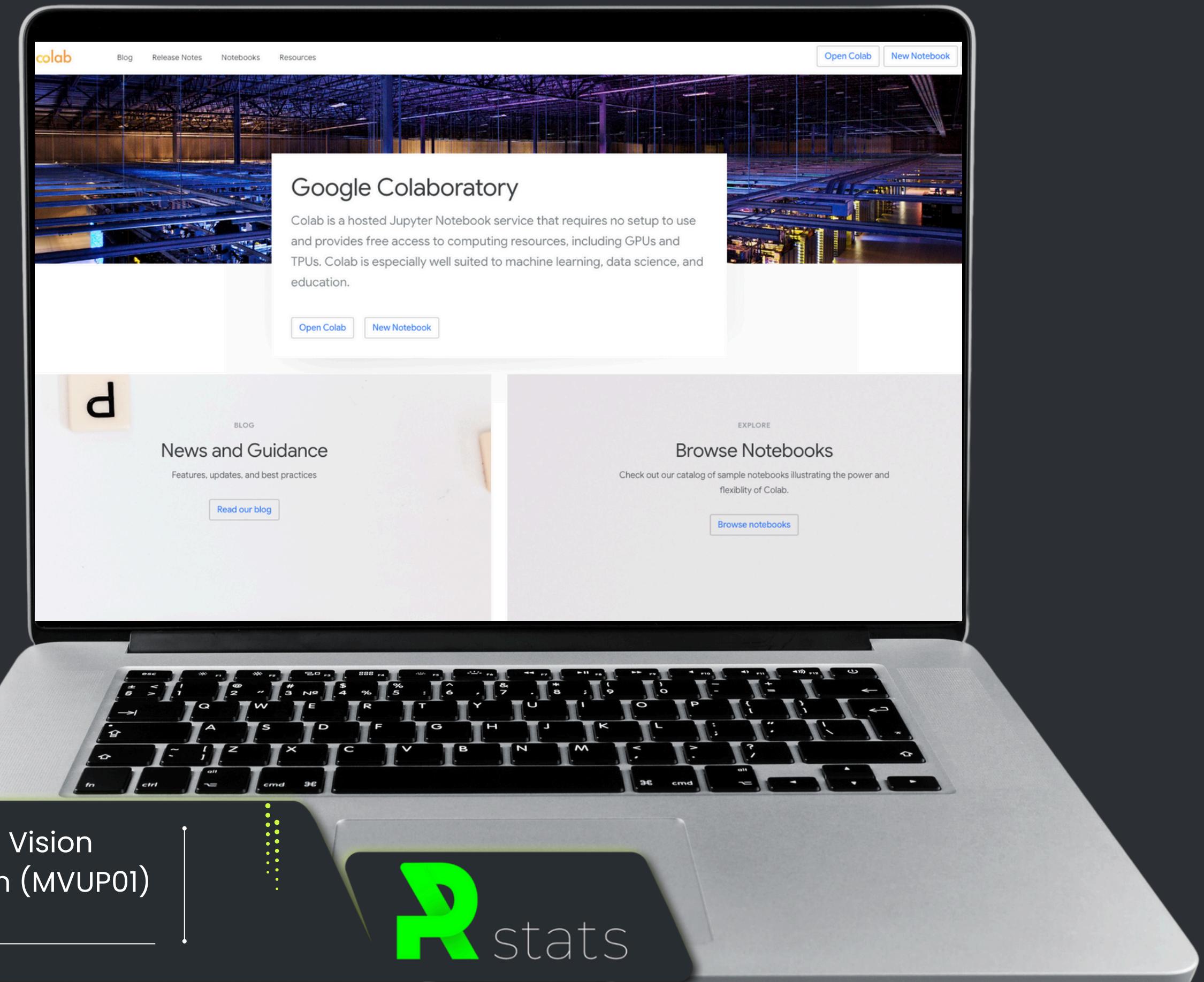
# Combining multiple conditions with logical operators
if (elevation >= 1000 and elevation <= 3500) and \
(rainfall > 500 and rainfall < 2000):
    print("GOOD: Elevation and rainfall within suitable range")
else:
    print("WARNING: Location outside preferred elevation or rainfall range")
```

Habitat Suitability Analysis for Mountain Gorillas:
GOOD: Elevation and rainfall within suitable range



Making Decisions in Code

Hands-on activity:



Machine Vision
using Python (MVUP01)

R stats

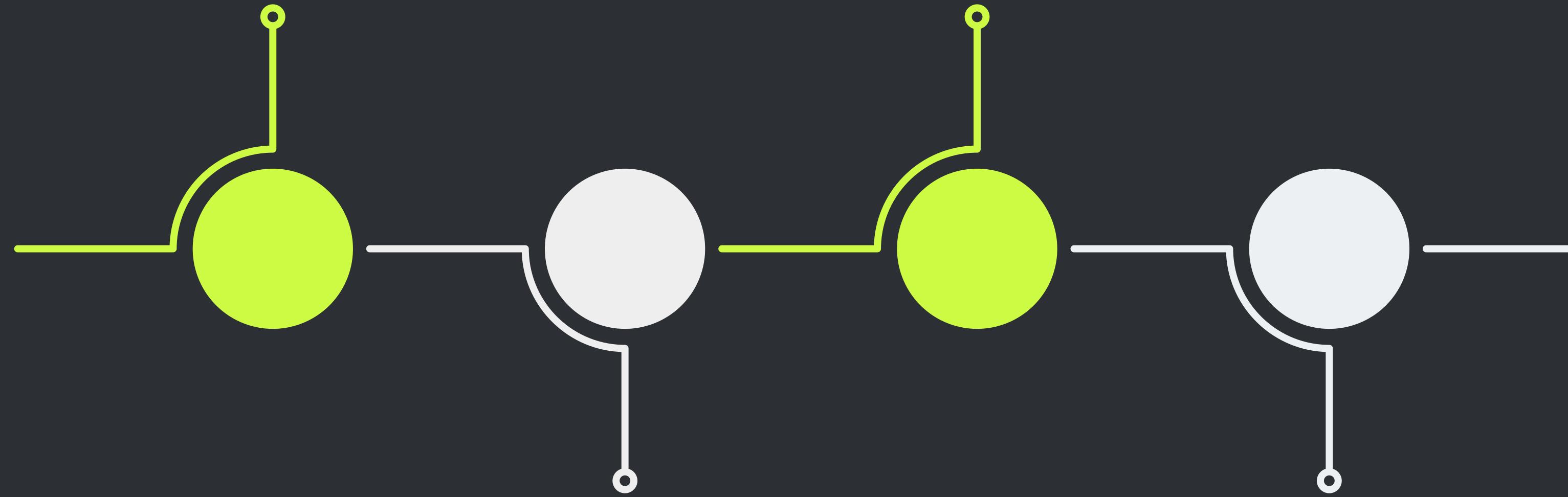
01001001010001
101001001010001
1001001010011100
0100100101000101
01001010001101
1010101000110011
1010101010110011
1010101010111111



Morning Session (9:30 - 12:00)

Your First Steps into
Python (9:45 - 10:00)

Making Decisions in
Code (10:45 - 11:15)



Machine Vision
using Python (MVUP01)



The Power of Repetition

Machine Vision
using Python (MVUP01)



The Power of Repetition



Machine Vision
using Python (MVUP01)



The Power of Repetition



Machine Vision
using Python (MVUP01)



The Power of Repetition

For loop implementation

```
# Number of sampling sites
total_sites = 2

print("Field Survey Data Collection:")
for site in range(1, total_sites + 1):
    print(f"\nSite {site} Analysis:")

    # Simulate measurements for each site
    temperature = 20 + site # Temperature increases with each site
    humidity = 60 - (site * 2) # Humidity decreases with each site

    # Analyze conditions for each site
    if temperature > 23:
        print(f"Temperature: {temperature}°C - Above optimal range")
    else:
        print(f"Temperature: {temperature}°C - Within optimal range")

    if humidity < 55:
        print(f"Humidity: {humidity}% - Below optimal range")
    else:
        print(f"Humidity: {humidity}% - Within optimal range")
```

Machine Vision
using Python (MVUP01)



The Power of Repetition

If/Elif/else structures

Field Survey Data Collection:

Site 1 Analysis:

Temperature: 21°C – Within optimal range

Humidity: 58% – Within optimal range

Site 2 Analysis:

Temperature: 22°C – Within optimal range

Humidity: 56% – Within optimal range

Machine Vision
using Python (MVUP01)



The Power of Repetition

While loop usage

```
# Number of sampling sites
total_sites = 2
current_site = 1

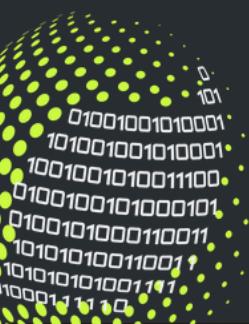
print("Field Survey Data Collection:")
while current_site <= total_sites:
    print(f"\nSite {current_site} Analysis:")

    # Simulate measurements for each site
    temperature = 20 + current_site # Temperature increases with each site
    humidity = 60 - (current_site * 2) # Humidity decreases with each site

    # Analyze conditions for each site
    if temperature > 23:
        print(f"Temperature: {temperature}°C - Above optimal range")
    else:
        print(f"Temperature: {temperature}°C - Within optimal range")

    if humidity < 55:
        print(f"Humidity: {humidity}% - Below optimal range")
    else:
        print(f"Humidity: {humidity}% - Within optimal range")

    current_site += 1
```



Machine Vision
using Python (MVUP01)



The Power of Repetition

While loop usage

Field Survey Data Collection:

Site 1 Analysis:

Temperature: 21°C – Within optimal range

Humidity: 58% – Within optimal range

Site 2 Analysis:

Temperature: 22°C – Within optimal range

Humidity: 56% – Within optimal range

Machine Vision
using Python (MVUP01)



The Power of Repetition

Loop control mechanisms

```
# Monitor daily temperature readings
max_temp = 35 # Maximum temperature threshold in Celsius
day = 1

print("Temperature Monitoring:")
while day <= 3: # Five days monitoring
    # Simulate daily temperature
    temperature = 30 + (day * 2) # Temperature increases each day

    print(f"\nDay {day}:")
    print(f"Temperature: {temperature}°C")

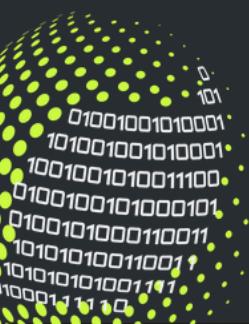
    # Skip recording if equipment maintenance (day 3)
    if day == 2:
        print("Equipment maintenance - no recording")
        day += 1
        continue

    # Check if temperature exceeds threshold
    if temperature > max_temp:
        print("WARNING: Temperature threshold exceeded!")
        break

    day += 1

print(f"\nMonitoring ended on day {day}")
```

Machine Vision
using Python (MVUP01)



The Power of Repetition

While loop usage

Temperature Monitoring:

Day 1:

Temperature: 32°C

Day 2:

Temperature: 34°C

Equipment maintenance – no recording

Day 3:

Temperature: 36°C

WARNING: Temperature threshold exceeded!

Monitoring ended on day 3

Machine Vision
using Python (MVUP01)





Lunch time

Machine Vision
using Python (MVUP01)

