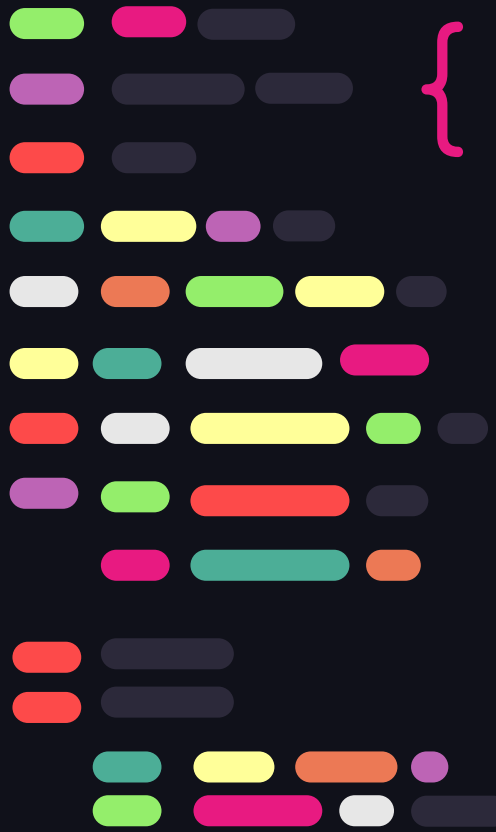


Introduction to Python

Resources

Useful Links

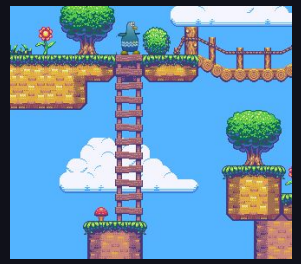
- https://docs.python.org/3/tutorial/index.html - Python Tutorial
- https://bit.ly/3JkAfa6 - Link to Slides
- https://forms.gle/m8tX4U5JtFDV71k99 - Question Form

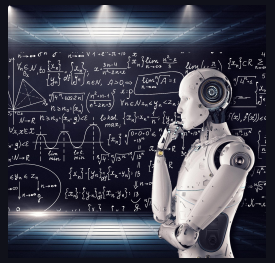
What is Python?

- Programming language
- Used to create programs and software we use

Pros:

- Easy to learn & read
- Widely used



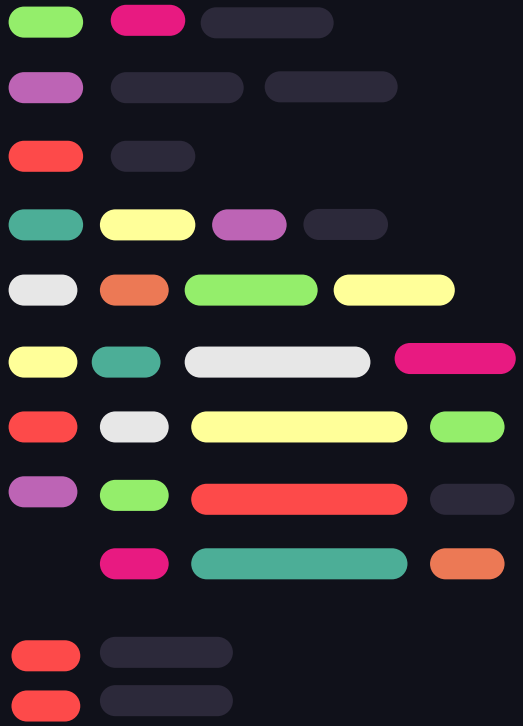


01 { ..

Setup

} ..

Our code editor: Replit!



For this activity, we will use repl.it

1. Go to replit.com
2. Click Sign up in the top right corner
3. Click Continue with Google
4. Sign in with your SFUSD email

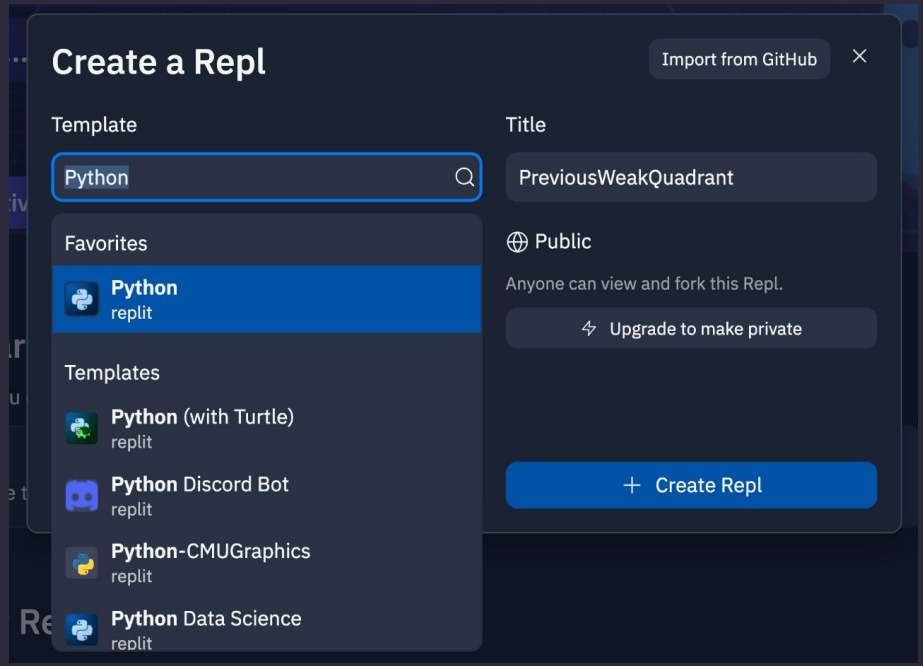
Remember: Ask questions if you are stuck!

Creating Your First Repl

This is how you create a new project:

1. Click Create Repl in the top left corner
2. In the template search box, type Python
3. Click Create Repl in the bottom right of the popup

You should be redirected to your project



02 { ..

The Basics

} ..

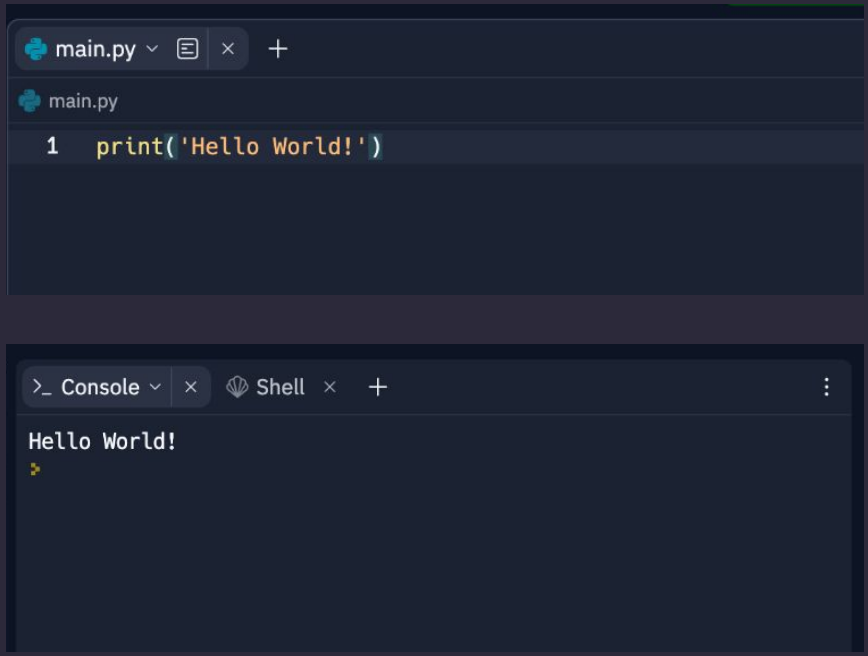
Your First Line of Code

Copy the text into the left text box.

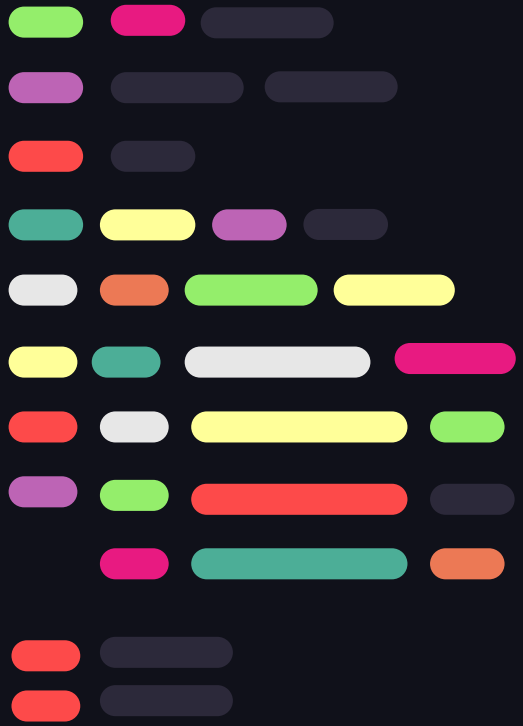
Click the green run button

Try changing the text inside the quotes.

Note: It is very important to copy the code and understand the code when learning Python



What does print do?



- The print function writes text into the console
- The console is the text output of your program

Next: We will learn about variables

Variables

- Names that hold some value
- <name> = <value>
- Can be used as value of print statements

main.py

```
1  name = 'John'
2  age = 21
3  print(name)
4  print(age)
```

Console × Shell × +

```
John
21
```

3 basic variable types

Number (int)

Stores a number

```
age = 10
```

Text (string)

Stores text

```
name = 'John'
```

Boolean (bool)

True / False

```
student = False
```

Values can be printed to console

Operations With Numbers

- + → add
- - → subtract
- * → multiply
- / → divide
- % → remainder
    - Ex: 11 % 2 = 1

```
1  a = 2
2  b = 3
3  c = a + b
4  print(c)
5  print(a + b)
```

```
5
5
```

Operations With Text (Strings)

- `+` → concatenation

```
1  first_name = 'John'
2  last_name = 'Doe'
3  full_name = first_name + ' ' + last_name
4  print(full_name)
5  print(first_name + last_name)
6  print(first_name + ' ' + last_name)
```

Output:

```
John Doe
JohnDoe
John Doe
```

More advanced Operators

Want to store a result into the same var?

`+=` → adds to a variable, stores result in same variable

```
1  number = 7
2  print(number)
3  number = number * 2
4  print(number)
5  number -= 4
6  print(number)
7
```

Output:
7
14
10

```
1  name = 'John'
2  print(name)
3  name += 'Joe'
4  print(name)
5  name += 'Jay'
6  print(name)
7
```

Output:
John
JohnJoe
JohnJoeJay

Reading Input

Use the input function to read text

Syntax:
`text = input(<prompt>)`

Now you (the player) can interact with the console!!!

main.py

```
1  name = input('Enter your name: ')
2  print('Your name is ' + name)
```

```
Enter your name: Simon
Your name is Simon
```

Input Dangers

`input()` always gives us a `string` (text) value type

To get value as an `int` (number), you have to `cast the value`

`int(<value>)` - changes value to a int (number) type

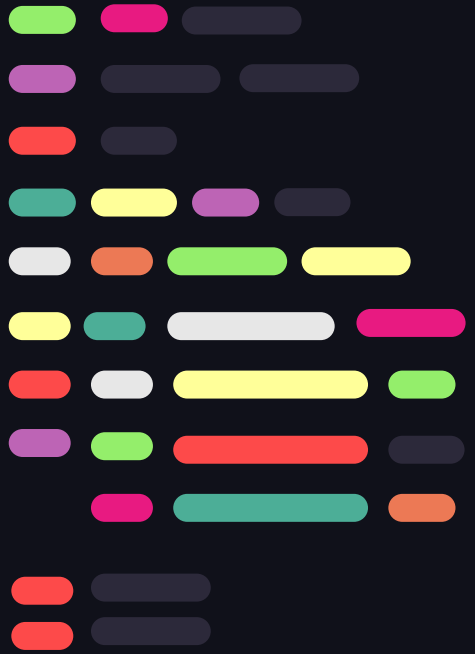
```
1  age = input('Enter your age: ')
2  age += 1
3  print('Your age is ' + age)
4
```

```
Enter your age: 10
Traceback (most recent call last):
  File "main.py", line 2, in <module>
    age += 1
TypeError: can only concatenate str (not "int") to str
```

```
1  age = int(input('Enter your age: '))
2  age += 1
3  print('Your age is ' + str(age))
4
```

```
Enter your age: 10
Your age is 11
```

Practical exercise 01



Write a program that allows the user to enter two numbers. Then, multiply the result and print to the console.

Example Output

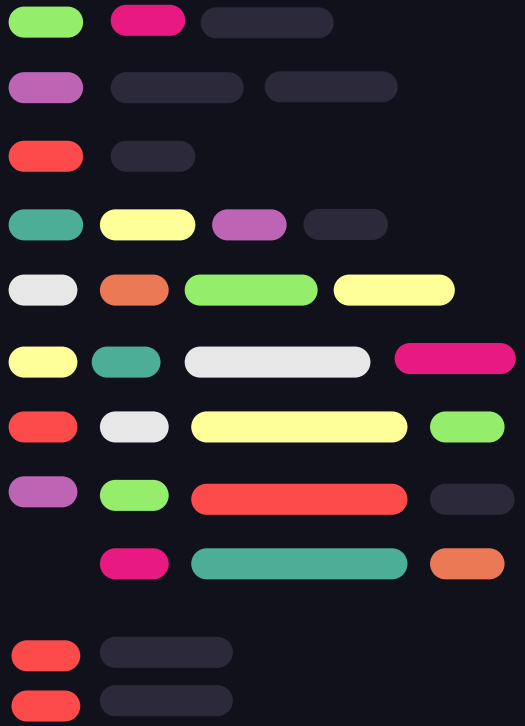
```
Enter the first number: 12
Enter the second number: 5
12 multiplied by 5 is: 60
```

03 { ..

Basic Data Structures

} ..

What is a data structure?



A data structure is a way to organize data

Ex: Shopping List

We will learn about three data structures

1. List
2. Tuple
3. Dictionary

Lists

An ordered sequence of elements

Uses brackets - []

Ex: Store a list of bullets

Danger: The list is zero indexed

Note: Lists can store lists!

```
1  nums = [1, 2, 3, 4, 5]
2  print(nums)
3  print(nums[0])
4  nums[0] = 10
5  print(nums)
6  print(nums[0])
```

Output

```
[1, 2, 3, 4, 5]
1
[10, 2, 3, 4, 5]
10
```

List Operations

append(<data>) → add to end of list

pop(<index>) → remove data at index <index>

insert(<index>, <data>) → add <data> at index <index>

sort() → sorts the data from least to greatest

len(<list>) → gets # items in <list>

```
1   nums = [1, 3, 2, 5, 4]
2   print(nums)
3   nums.append(6)
4   print(nums)
5   nums.insert(1, 7)
6   print(nums)
7   nums.pop(2)
8   print(nums)
9   nums.sort()
10  print(nums)
```

Output

```
[1, 3, 2, 5, 4]
[1, 3, 2, 5, 4, 6]
[1, 7, 3, 2, 5, 4, 6]
[1, 7, 2, 5, 4, 6]
[1, 2, 4, 5, 6, 7]
```

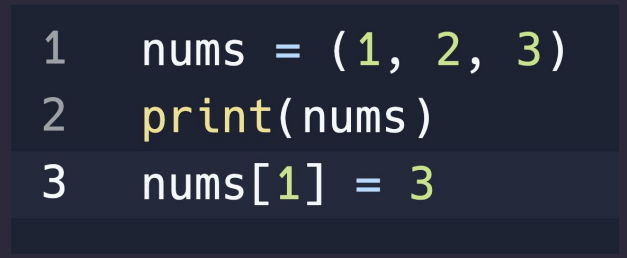
Tuples

A sequence of elements that can't be modified

Uses parentheses - ()

List operations don't work on tuples

Pygame uses tuples



```
1   nums = (1, 2, 3)
2   print(nums)
3   nums[1] = 3
```

```
(1, 2, 3)
Traceback (most recent call last):
  File "main.py", line 3, in <module>
    nums[1] = 3
TypeError: 'tuple' object does not support item assignment
```

Dictionaries

A list of key value pairs

Uses curly braces - {}

key - must be a string
value - can be any type

```
1  ages = {'John': 15, 'Jane': 12}
2  print(ages)
3  ages['Joe'] = 13
4  print(ages)
5  print(ages['John'])
```

Output

```
{‘John’: 15, ‘Jane’: 12}
{‘John’: 15, ‘Jane’: 12, ‘Joe’: 13}
15
```

Dictionary Operations

pop(<key>) → deletes entry with key <key>

Not very important:
keys() → get the list of keys in the dictionary

values() → get the list of values in the dictionary

```python
nicknames = {
    'Alexander': 'Alex',
    'Jonathan': 'Johnny',
    'Nicholas': 'Nick',
}
print(nicknames)
print(nicknames.keys())
print(nicknames.values())
nicknames.pop('Alexander')
print(nicknames)
```

```
{'Alexander': 'Alex', 'Jonathan': 'Johnny', 'Nicholas': 'Nick'}
dict_keys(['Alexander', 'Jonathan', 'Nicholas'])
dict_values(['Alex', 'Johnny', 'Nick'])
{'Jonathan': 'Johnny', 'Nicholas': 'Nick'}
```

04 { ..

Control Flow

} ..

Boolean Operators

Result is either True or False

< → Less than
<= → Less than or equal
== → Equal
> → Greater than
>= → Greater than or equal

```
1  num_1 = 10
2  num_2 = 5
3  num_3 = 12
4
5  print(num_1 > num_2)
6  print(num_3 < num_2)
7  print(num_2 + 6 == num_1)
8  print(num_2 + 6 >= num_1)
9  print(num_3 - 10 <= num_3)
```

Output
True
False
False
True
True

Logic Operators

<bool> and <bool2> → True if both <bool> and <bool2> is True

<bool> or <bool2> → True if either <bool> or <bool2> is true

not <bool> → Returns the opposite of <bool>

```
1   bool_1 = True
2   bool_2 = False
3   bool_3 = True
4
5   print(bool_1 and bool_2)
6   print(bool_1 and bool_3)
7   print(bool_1 or bool_2)
8   print(bool_1 or bool_3)
9   print(not bool_1)
10  print(not bool_2)
```

Output
```
False
True
True
True
False
True
```

If statement

Runs indented code if the condition is True

```
if <bool>:
    more code ...
```

elif → runs if previous condition is False

else → runs if all conditions are False

```
1  age = 15
2  if age <= 10:
3      print('You are in elementary school')
4  elif age > 10 and age < 14:
5      print('You are in middle school')
6  elif age <= 18:
7      print('You are in high school')
8  else:
9      print('You are in college')
```

Output

You are in high school

While Loops

```
while <bool>:
    Code ...
```

Runs indented code until <bool> becomes false

Danger: Infinite Loops

Your program will run forever

```
1  i = 1
2  while i < 6:
3      print(i)
4      i += 1
```

Output

```
1
2
3
4
5
```

A Practical Example

```python
nums = [5, 10, 15, 20, 25]
i = 0
while i < len(nums):
    print(nums[i])
    i += 1
```

Output:

```
5
10
15
20
25
```

{

Using a while loop to print every element from an array called nums

}

Practical exercise 02

```
Guess the number: 500
Guess is too low
Guess the number: 750
Guess is too low
Guess the number: 875
Guess is too low
Guess the number: 925
Guess is too low
Guess the number: 950
Guess is too low
Guess the number: 975
Guess is too low
Guess the number: 987
Guess is too low
Guess the number: 993
Guess is too high
Guess the number: 990
Guess is too low
Guess the number: 992
You win! The number was 992
```

Create a number guessing game.

1. Tell the player if they guess too low, guess too high, or guess correctly

2. The game will keep running until the user guesses the number correctly.

Hint: To generate a random target number for the game, use `randint()`

`randint(a, b)` → generates a random number between `a` and `b` inclusive

```
1  from random import randint
2  answer = randint(1, 1000)
```

For Each Loops

Repeats for each item in list

num stores the current number in the list

Variable after “in” must be a list

```
1  nums = [25, 5, 15, 20, 10]
2  for num in nums:
3      print(num)
```

Output
25
5
15
20
10

For Loops

Runs indented code until for a specified number of times

range(<start>, <stop>) → generates a list of numbers from <start>, ending before <stop>, + 1 each time

range(a, b, <inc>) → same as above but <inc> can specify the increase value each time

```
1  i = 1
2  while i < 6:
3      print(i)
4      i += 1
```

```
1  for i in range(1, 6):
2      print(i)
```

range(1, 6) = [1, 2, 3, 4, 5]

Output
1
2
3
4
5

The Three Loops

While

```
1  nums = [25, 5, 15, 20, 10]
2  i = 0
3  while i < len(nums):
4      print(nums[i])
5      i += 1
```

For

```
1  nums = [25, 5, 15, 20, 10]
2  for i in range(len(nums)):
3      print(nums[i])
```

For Each

```
1  nums = [25, 5, 15, 20, 10]
2  for num in nums:
3      print(num)
```

Try Except Blocks

Runs catch block when try block throws an error

Used to handle errors with messages

```python
is_num = False

while not is_num:
  try:
    num = int(input('Enter a number: '))
    is_num = True
    print('You have entered a number: ' + str(num))
  except:
    print('Invalid entry. Try again.')
```

```
Enter a number: hello
Invalid entry. Try again.
Enter a number: wrong
Invalid entry. Try again.
Enter a number: thing
Invalid entry. Try again.
Enter a number: 32
You have entered a number: 32
```

Functions

Used to reduce duplicate code

Ex: Making a cake. The steps are hidden

nums is called an argument

arguments are data that you pass into a function

Functions can have >= 0 arguments

```python
def print_nums(nums):
    for num in nums:
        print(num)
    print(nums)
    print()

print_nums([1, 2, 3, 4, 5])
print_nums([6, 23, 22, 0, 9])
```

```
1
2
3
4
5
[1, 2, 3, 4, 5]

6
23
22
0
9
[6, 23, 22, 0, 9]
```

Return Values

When you call a function, you can return the result back to the caller

```
1  def add(a, b):
2      return a + b
3
4  total = add(10, 15)
5  print(total)
```

Output
25

Practical exercise 03

```
Make a choice (R - Rock, P - Paper, S - Scissors): R
You chose R and the bot chose S
You win!
Continue playing? (y/n): y
Make a choice (R - Rock, P - Paper, S - Scissors): sjfas
Invalid choice: sjfas
Make a choice (R - Rock, P - Paper, S - Scissors): P
You chose P and the bot chose S
You lose!
Continue playing? (y/n): n
```

```python
from random import choice
nums = [1, 2, 3, 4]
print(choice(nums))
```

Create a Rock, Paper, Scissors Game.

Allow the user to keep playing until the decides to quit

Topics: Inputs, Loops, Strings, If

Hint:
choice(<list>) → Returns a random item in <list>

05 { ..

Pygame

} ..

Good Job!

Good job for learning as much as possible in this difficult topic

Here is the link to the pygame documentation:

https://www.pygame.org/docs/ - Pygame Documentation

Note: If you understand the previous slides, this should not be any more difficult

What is Pygame?

Pygame is a Python module that allows you to create games

There are two main components of Pygame:

- The Game Loop
- The Event Loop

Try out this game:

https://replit.com/@SimonZhao9/Snake

```python
import pygame


pygame.init()
screen = pygame.display.set_mode((400, 300))
pygame.display.set_caption('Hello World!')
game_over = False
white = (255, 255, 255)

# Game Loop
while not game_over:
    # Event Loop
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_over = True
    screen.fill(white)
    pygame.display.flip()
```

Initializing the Module

`pygame.init()` → initializes all pygame modules

`set_mode((width, height))` → Creates a window that is of the specified dimensions

`pygame.time.Clock()` → creates and returns a clock

Try out this game:

https://replit.com/@SimonZhao9/Snake

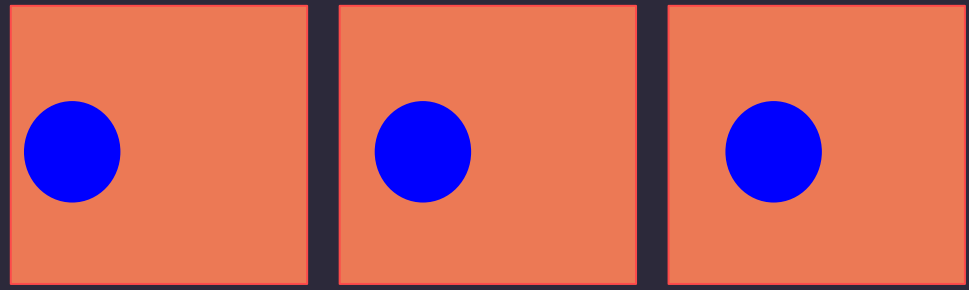
```
1  import pygame
2
3
4  pygame.init()
5  screen = pygame.display.set_mode((400, 300))
6  pygame.display.set_caption('Hello World!')
7  game_over = False
8  white = (255, 255, 255)
9
10 # Game Loop
11 while not game_over:
12     # Event Loop
13     for event in pygame.event.get():
14         if event.type == pygame.QUIT:
15             game_over = True
16     screen.fill(white)
17     pygame.display.flip()
```

The Game Loop

The game loop is the main (outer) while loop of your game

Idea: You redraw elements on the screen to make it seem like the object moved

So: Every loop, you calculate a new position, then redraw your screen



```python
import pygame


pygame.init()
screen = pygame.display.set_mode((400, 300))
pygame.display.set_caption('Hello World!')
game_over = False
white = (255, 255, 255)

# Game Loop
while not game_over:
    # Event Loop
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_over = True
    screen.fill(white)
    pygame.display.flip()
```

The Event Loop

The event loop is an inner while loop

Checks for all game events

Ex: Mouse click, Button Press, Quit

```python
import pygame


pygame.init()
screen = pygame.display.set_mode((400, 300))
pygame.display.set_caption('Hello World!')
game_over = False
white = (255, 255, 255)

# Game Loop
while not game_over:
    # Event Loop
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_over = True
    screen.fill(white)
    pygame.display.flip()
```

Surfaces

Surfaces are areas where you place content on

The set_mode(...) function creates a surface so you can draw content on it

<surface>.fill((red, green, blue)) → fill the surface with the specified RGB color

```python
import pygame


pygame.init()
screen = pygame.display.set_mode((400, 300))
pygame.display.set_caption('Hello World!')
game_over = False
white = (255, 255, 255)

# Game Loop
while not game_over:
    # Event Loop
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_over = True
    screen.fill(white)
    pygame.display.flip()
```

Coordinates

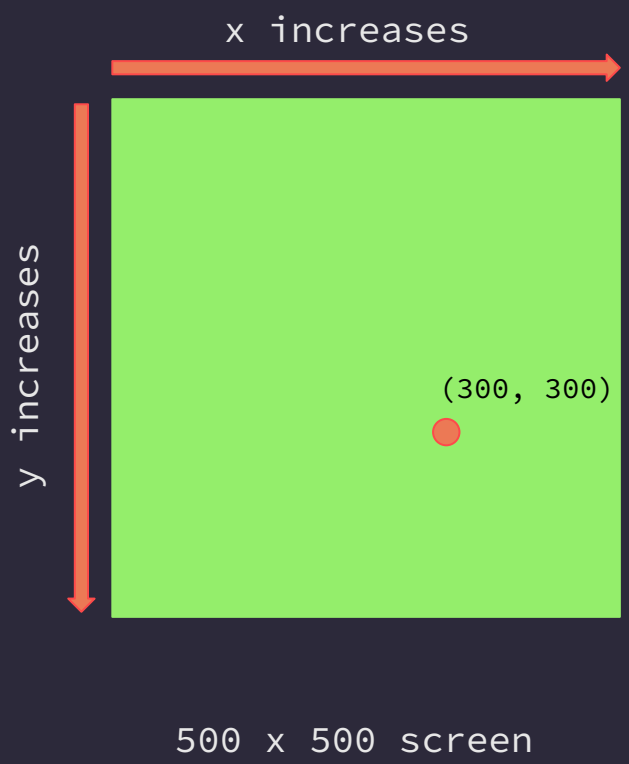
x - represents horizontal position

y - represents vertical position

As y increases, position goes down

(<x>, <y>) - represents a position

The position: (0, 0) is the top-left corner of screen



Rect

A rectangle in pygame.
Used for positioning

pygame.rect.Rect(x, y, width, height) → creates a rectangle with arguments

x, y is for rect's top-left corner

pygame.draw.rect(surface, color, rect) → Draws the rectangle onto the surface with color

```python
import pygame


pygame.init()
screen = pygame.display.set_mode((400, 300))
pygame.display.set_caption('Hello World!')
game_over = False

# Colors
white = (255, 255, 255)
red = (255, 0, 0)

cube = pygame.rect.Rect(200, 200, 50, 50)

# Game Loop
while not game_over:
    # Event Loop
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_over = True
    screen.fill(white)
    pygame.draw.rect(screen, red, cube)
    pygame.display.flip()
```

Images

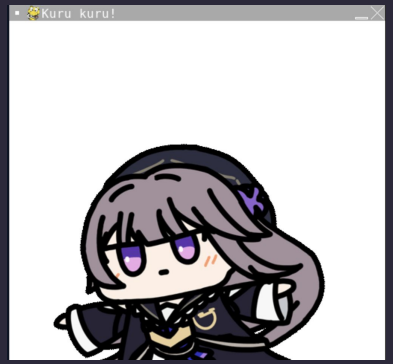
3 Steps: Load, position, show image

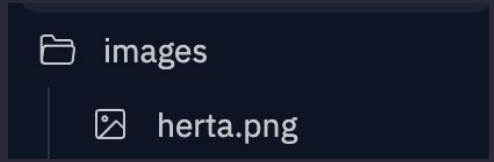
`pygame.image.load(filepath)` → loads image and returns a surface

`<surface>.get_rect()` → returns a Rect used for positioning

`<surface>.blit(image, rect)` → draws the `image` on top of `<surface>`, positioned the same way as `rect`

```
14  screen_rect = screen.get_rect()
15  image = pygame.image.load('images/herta.png')
16  image_rect = image.get_rect()
17  image_rect.bottom = screen_rect.bottom
18  image_rect.centerx = screen_rect.centerx
19
20  # Game Loop
21  while not game_over:
22      # Event Loop
23      for event in pygame.event.get():
24          if event.type == pygame.QUIT:
25              game_over = True
26      screen.fill(white)
27      pygame.draw.rect(screen, red, cube)
28      screen.blit(image, image_rect)
29      pygame.display.flip()
30
```





← Result

Full Code

Just in case you need it...

```python
import pygame

pygame.init()
screen = pygame.display.set_mode((500, 500))
pygame.display.set_caption('Kuru kuru!')
game_over = False

# Colors
white = (255, 255, 255)
red = (255, 0, 0)

cube = pygame.rect.Rect(200, 200, 50, 50)

screen_rect = screen.get_rect()
image = pygame.image.load('images/herta.png')
image_rect = image.get_rect()
image_rect.bottom = screen_rect.bottom
image_rect.centerx = screen_rect.centerx

# Game Loop
while not game_over:
    # Event Loop
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_over = True
    screen.fill(white)
    pygame.draw.rect(screen, red, cube)
    screen.blit(image, image_rect)
    pygame.display.flip()

```

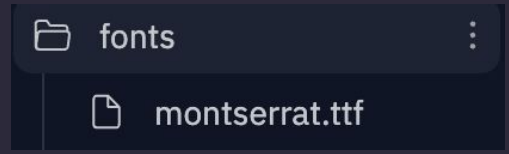
Text

Text is displayed as an image

pygame.font.Font(file, size) → creates a Font with specified font file (can be None) and font size

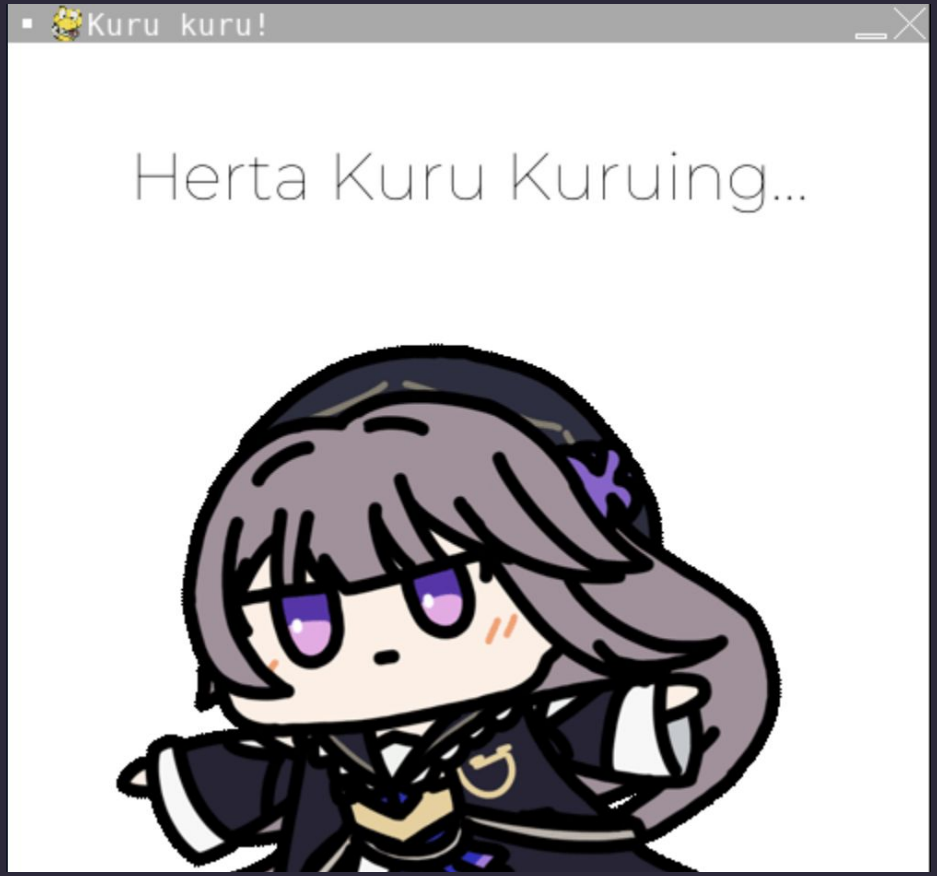
<font>.render(text, True, color) → Returns a image with the text and text color

```
21  font = pygame.font.Font('fonts/montserrat.ttf', 36)
22  header = font.render('Herta Kuru Kuruing...', True, black)
23  header_rect = header.get_rect()
24  header_rect.centerx = screen_rect.centerx
25  header_rect.y = 50
26
27  # Game Loop
28  while not game_over:
29      # Event Loop
30      for event in pygame.event.get():
31          if event.type == pygame.QUIT:
32              game_over = True
33      screen.fill(white)
34      pygame.draw.rect(screen, red, cube)
35      screen.blit(image, image_rect)
36      screen.blit(header, header_rect)
37      pygame.display.flip()
38
```



Text Output

After adding a text image onto the screen ...



Event Types

pygame.MOUSEBUTTONDOWN

When you left click with mouse

pygame.KEYDOWN

When a keyboard key goes down

pygame.KEYUP

When a keyboard key goes up

Note: Pressing a key once triggers KEYDOWN & KEYUP

Events

Use if statements in event loop to act based on event types

<event>.type → type of event

For KEYDOWN/KEYUP only:
<event>.key → the key pressed

pygame.K_<key> → the key
Ex: pygame.K_w → W key

```
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        game_over = True
    elif event.type == pygame.KEYDOWN:
      if event.key == pygame.K_w:
        image_rect.y -= speed
      elif event.key == pygame.K_a:
        image_rect.x -= speed
      elif event.key == pygame.K_s:
        image_rect.y += speed
      elif event.key == pygame.K_d:
        image_rect.x += speed
```

Contents of this template

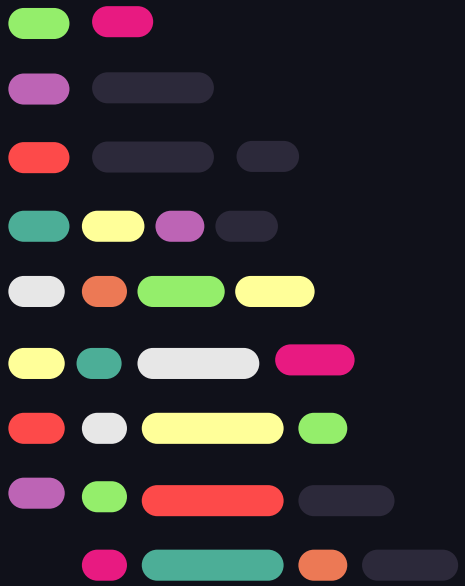
You can delete this slide when you're done editing the presentation

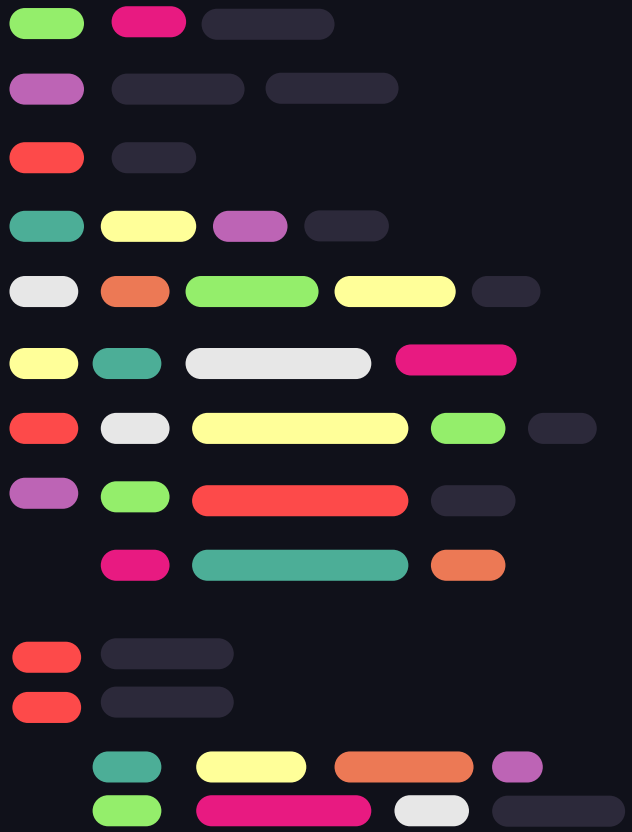
Fonts	To view this template correctly in PowerPoint, download and install the fonts we used
Used and alternative resources	An assortment of graphic resources that are suitable for use in this presentation
Thanks slide	You must keep it so that proper credits for our design are given
Colors	All the colors used in this presentation
Icons and infographic resources	These can be used in the template, and their size and color can be edited
Editable presentation theme	You can edit the master slides easily. For more info, click **here**

For more info:
SLIDESGO | BLOG | FAQS

You can visit our sister projects:
FREEPIK | FLATICON | STORYSET | WEPIK | VIDEVO

Table of contents





Whoa!}

< This can be the part of the presentation where you introduce yourself, write your email… >

01 { ..

Name of the section

< You can enter a subtitle here if you need it >

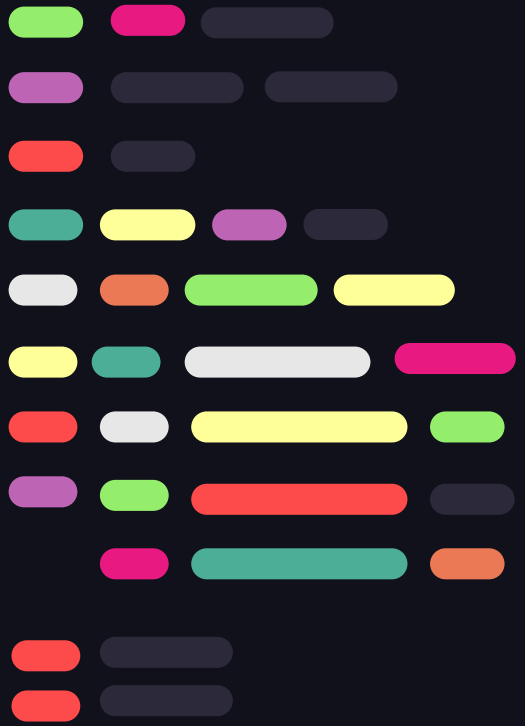
} ..

Do you need longer text?

Mercury is the closest planet to the Sun and the smallest one in the entire Solar System. This planet's name has nothing to do with the liquid metal, since Mercury was named after the Roman messenger god. It isn’t as hot as Venus, for instance

Mercury takes a little more than 58 days to complete its rotation, so try to imagine how long days must be there! Since the temperatures are so extreme, albeit not as extreme as on Venus, Mercury has been deemed to be non-habitable for humans

The slide title goes here!



Do you know what helps you make your point crystal clear? Lists like this one:

- They’re simple
- You can organize your ideas clearly
- You’ll never forget to buy milk!

And the most important thing: the audience won’t miss the point of your presentation

You can divide the content

Mercury

Mercury is the closest planet to the Sun and the smallest one in the Solar System—it's only a bit larger than the Moon

Venus

Venus has a beautiful name and is the second planet from the Sun. It's hot and has a poisonous atmosphere

Here are three ideas

Mercury

Mercury is the closest planet to the Sun and the smallest of them all

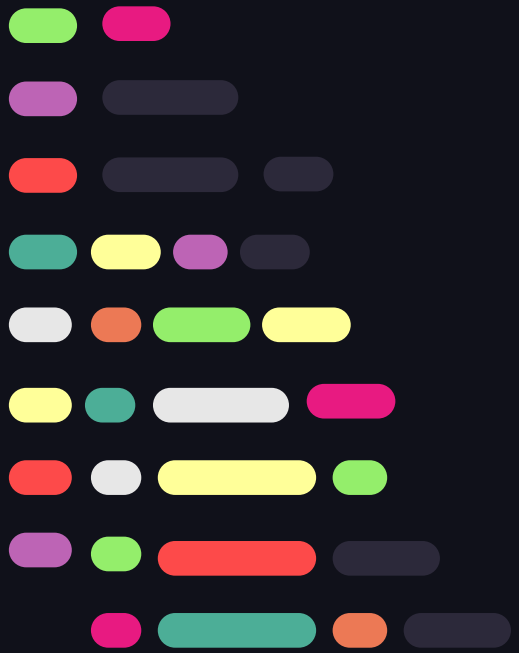
Venus

Venus has a beautiful name and is the second planet from the Sun

Mars

Despite being red, Mars is actually a cold place. It’s full of iron oxide

Practical exercise 01

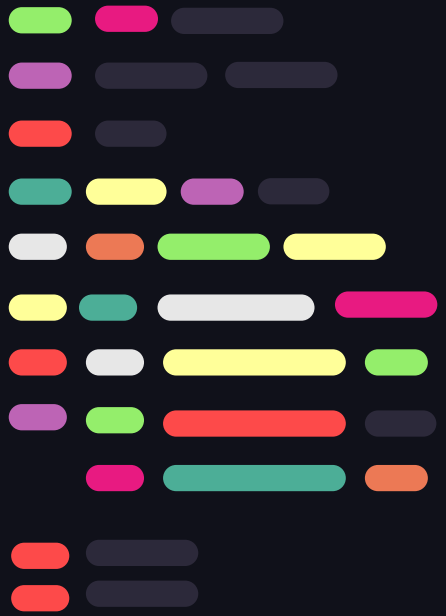


Write a Java program to print 'Hello' on screen and then print your name on a separate line.

Expected output:

```
Hello
Sofia Hill
```

Here are four concepts



Mars

Despite being red, Mars is a cold place

Venus

Venus has a nice name and a very toxic atmosphere

Jupiter

Jupiter is the biggest planet of them all

Saturn

Saturn is a gas giant and has several rings

{ .. Here are six concepts

Mars

Despite being red, Mars is a cold place

Venus

Venus has a nice name and a very toxic atmosphere

Neptune

Neptune is the farthest planet from the Sun

Mercury

Mercury is the closest planet to the Sun

Saturn

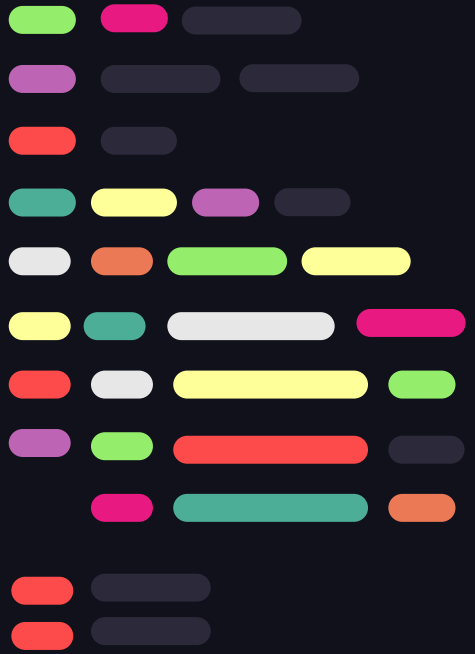
Saturn is a gas giant with several rings

Jupiter

Jupiter is the biggest planet of them all

} ..

Practical exercise 02



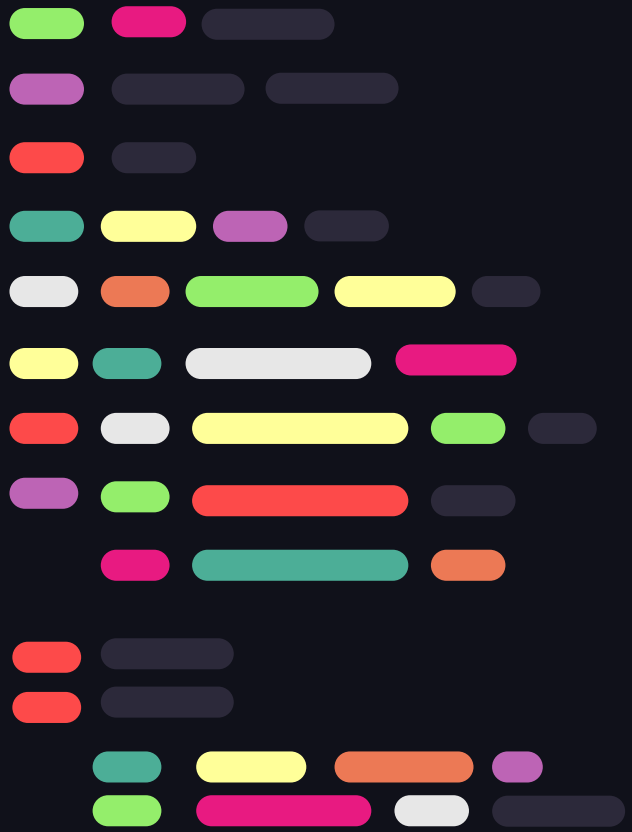
Write a Java program to divide two numbers and print on the screen.

Test data:

80/2

Expected output:

40

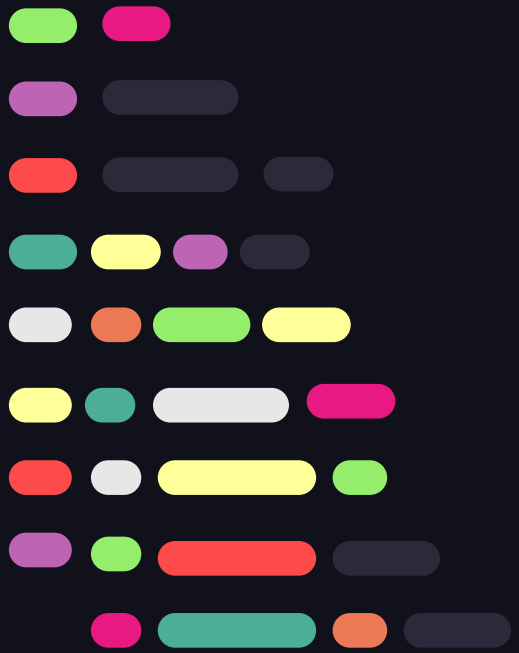


{ ..

Awesome words

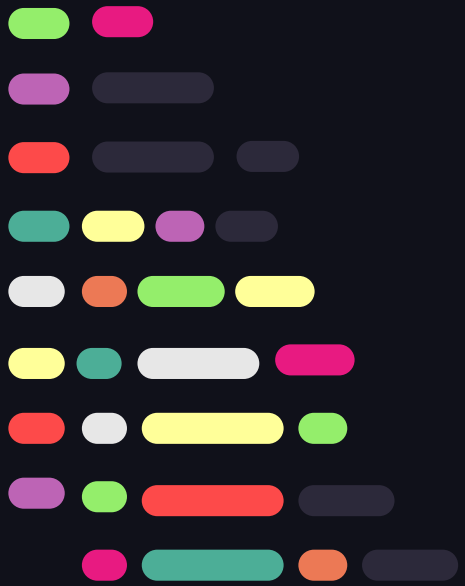
} ..

Practical exercise 03



Write a Java program that takes two numbers as input and displays the product of two numbers.

Test data:
Input first number: 70
Input second number: 4
Expected output:
70 x 4 = 280



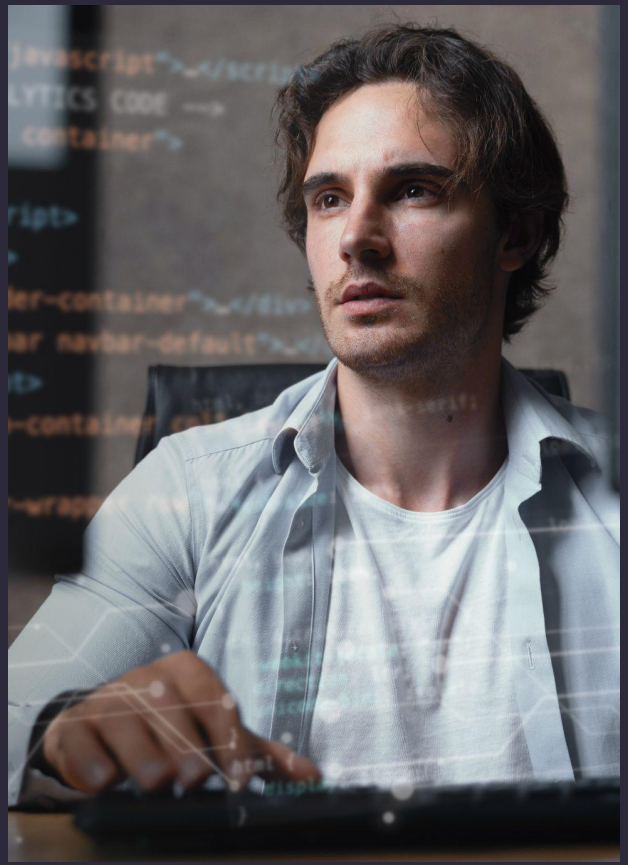
“This is a quote, words full of wisdom that someone important said and that can inspire the reader.”

—Someone Famous

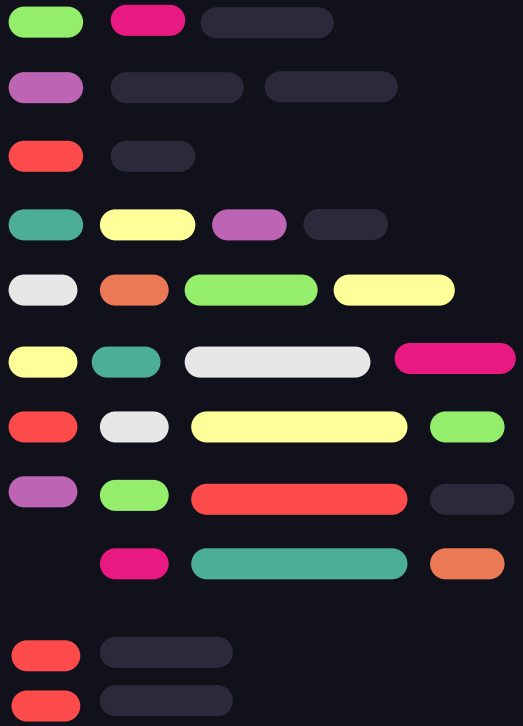

A picture is worth a thousand words

A picture always reinforces the concept

Images reveal large amounts of data, so remember: use an image instead of a long text. Your audience will appreciate it



Practical exercise 04



Write a Java program to print the sum (addition), multiplication, subtraction, division and remainder of two numbers.

Test data:

Input first number: 200

Input second number: 55

Expected Output:

200 + 55 = 255

200 - 55 = 145

200 x 55 = 11000

200 / 55 = 3,63

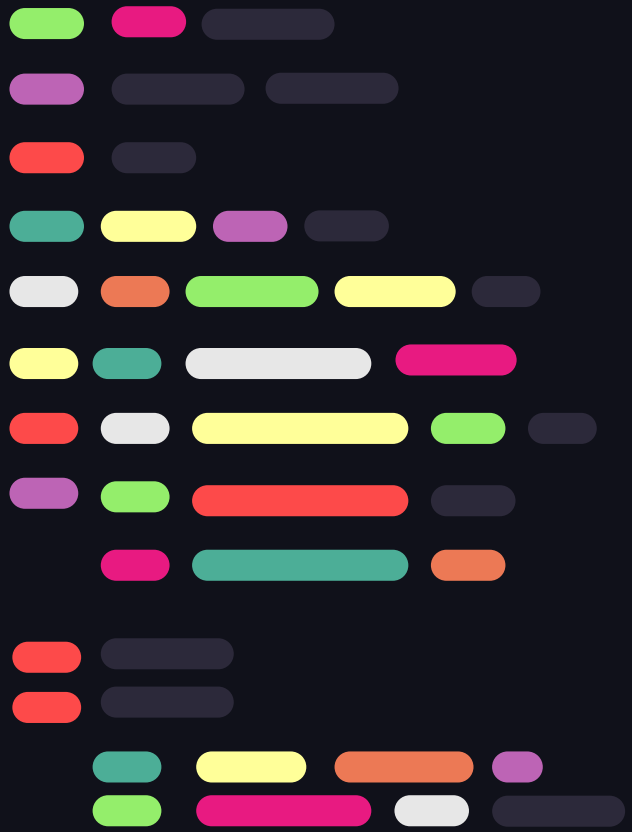
200 mod 5 = 35

{

98,300,000

< Big numbers catch your audience’s attention >

}



9h 55m 23s

Jupiter’s rotation period

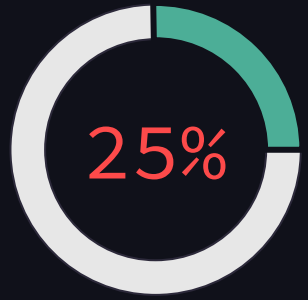
333,000

The Sun’s mass compared to Earth’s

386,000 km

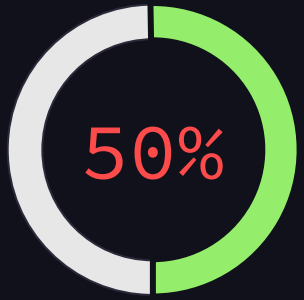
Distance between Earth and the Moon

Let’s use some percentages



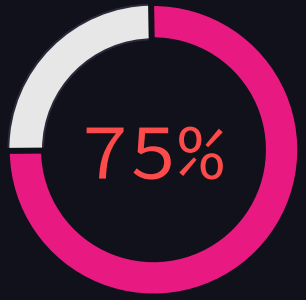
Mercury

Mercury is the closest planet to the Sun and the smallest one



Venus

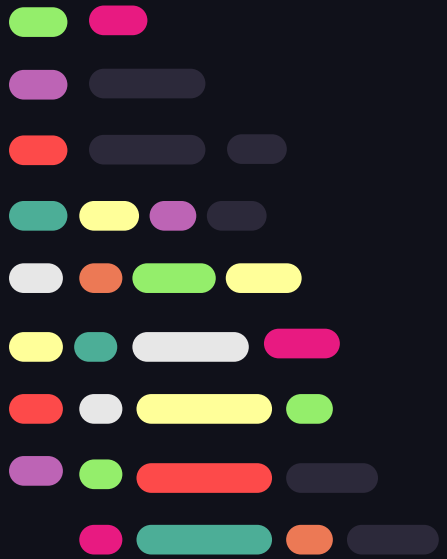
Venus has a beautiful name and is the second planet from the Sun



Mars

Despite being red, Mars is actually a cold place. It’s full of iron oxide

Practical exercise 05



Write a Java program to convert a binary number into a hexadecimal number.

Input data:

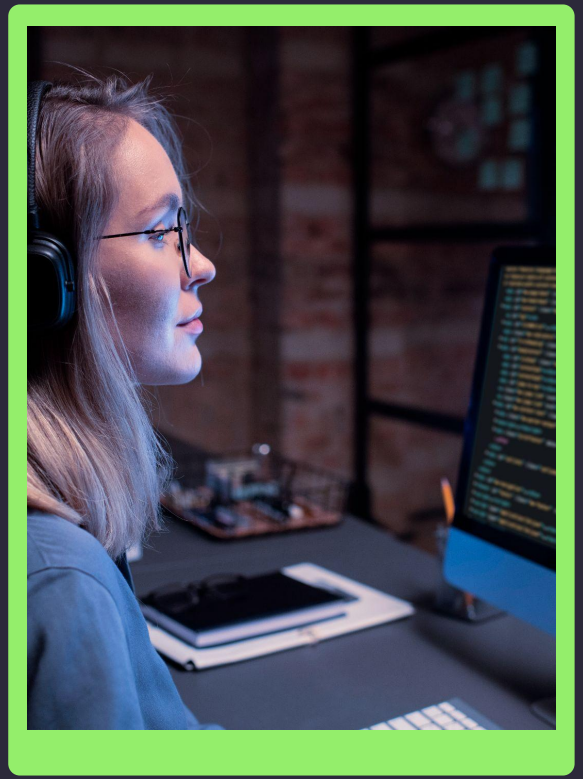
Input a binary number: 10010

Expected output:

Hexadecimal value: D

Computer mockup

You can replace the image on the screen with your own work. Just right-click on it and select “Replace image”



Tablet mockup

You can replace the image on the screen with your own work. Just right-click on it and select “Replace image”

Phone mockup

You can replace the image on the screen with your own work. Just right-click on it and select “Replace image”

This is a map

Venus

Venus is the second planet from the Sun

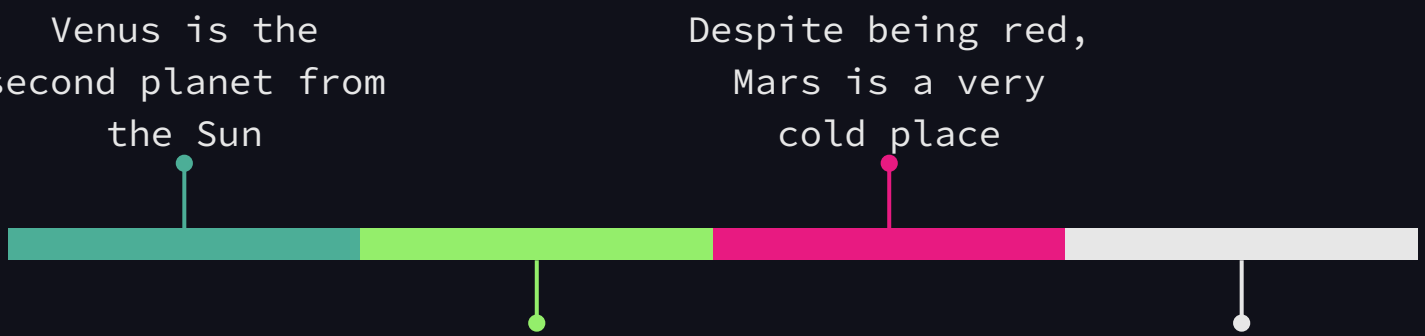
Mercury

Mercury is the closest planet to the Sun

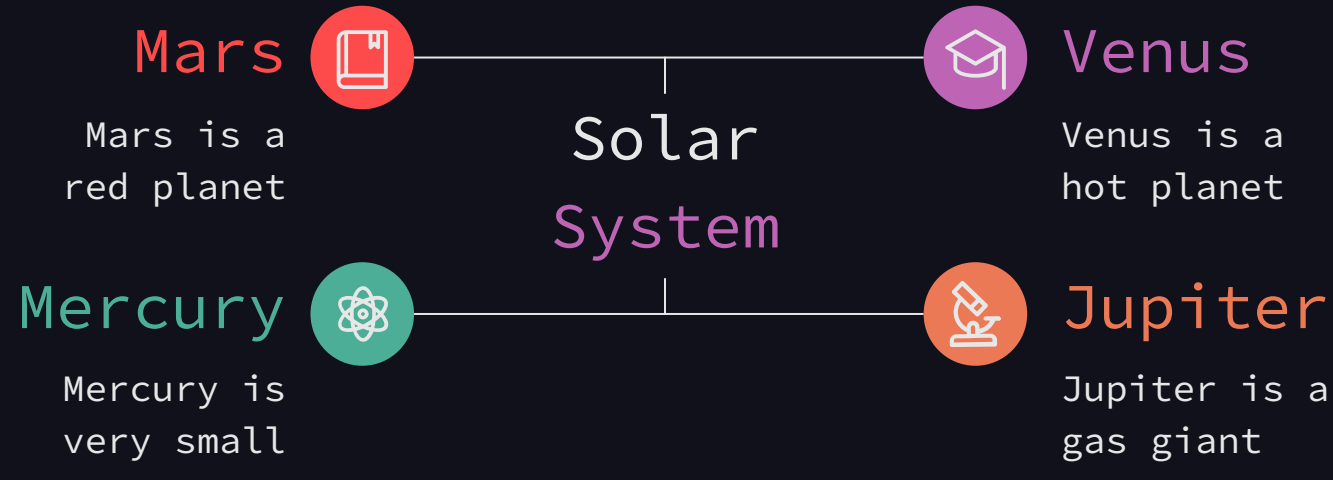
Mars

Despite being red, Mars is a very cold place

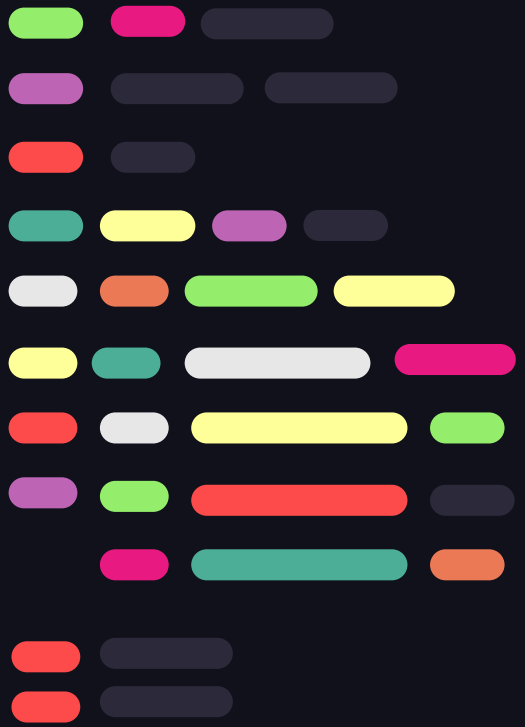
A timeline always works well



Infographics are useful



Practical exercise 06



Write a Java program to check whether Java is installed on your computer.

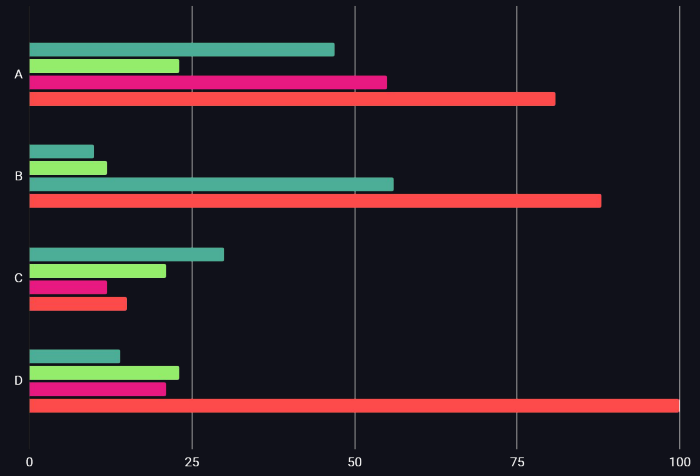
Expected output:

```
Java Version: 1.8.0_71
Java Runtime Version: 1.8.0_71-b15
Java Home: /opt/jdk/jdk1.8.0_71/jre
Java Vendor: Oracle Corporation
Java Vendor URL:
http://Java.oracle.com/
Java Class Path: .
```

Use tables to represent data

	Team A	Team B	Team C	Team D
Mercury	0.06	0.53	0.38	0.38
Mars	0.11	9.4	0.53	0.78
Saturn	95.2	1.16	9.4	1.16

You can use this graph



Mercury

Mercury is the smallest planet

Venus

Venus has very high temperatures

Jupiter

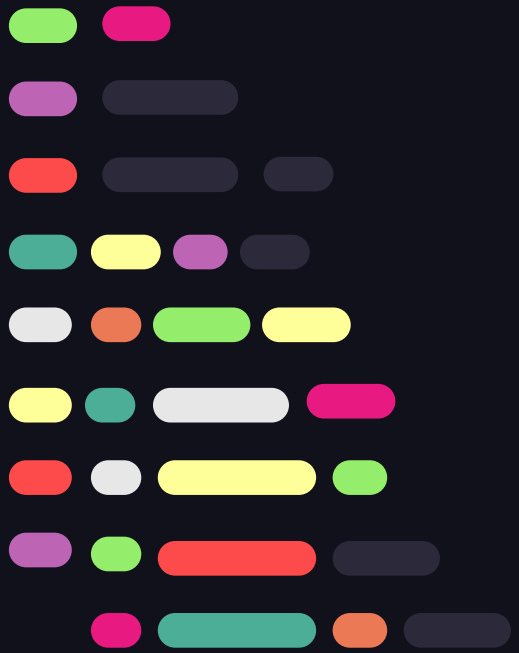
Jupiter is the biggest planet

Saturn

Saturn is a gas giant with rings

Follow the link in the graph to modify its data and then paste the new one here. **For more info, click here**

Practical exercise 07



Write a Java program to reverse a string.

Input data:

Input a string: My dog’s name is

Ralph

Expected output:

Reverse string: hplaR si eman s‘god

yM

Our team

Sofia Hill

You can speak a bit about this person here

Alex Harris

You can speak a bit about this person here

Thanks!

< Do you have any questions? >

youremail@freepik.com
+34 654 321 432
yourwebsite.com

Alternative resources

Here’s an assortment of alternative resources whose style fits the one of this template:

- Programmers concept with flat design

Resources

Did you like the resources on this template? Get them for free at our other websites:

Photos

- Html and css collage concept with person I
- Html and css collage concept with person II
- Html and css collage concept with person III
- Html and css collage concept with person IV
- Medium shot smiley woman at library
- Medium shot man holding notebook

Instructions for use

If you have a free account, in order to use this template, you must credit **Slidesgo** by keeping the **Thanks** slide. Please refer to the next slide to read the instructions for premium users.

As a Free user, you are allowed to:

- Modify this template.
- Use it for both personal and commercial projects.

You are not allowed to:

- Sublicense, sell or rent any of Slidesgo Content (or a modified version of Slidesgo Content).
- Distribute Slidesgo Content unless it has been expressly authorized by Slidesgo.
- Include Slidesgo Content in an online or offline database or file.
- Offer Slidesgo templates (or modified versions of Slidesgo templates) for download.
- Acquire the copyright of Slidesgo Content.

For more information about editing slides, please read our FAQs or visit our blog:
https://slidesgo.com/faqs and **https://slidesgo.com/slidesgo-school**

Instructions for use (premium users)

As a Premium user, you can use this template without attributing **Slidesgo** or keeping the **Thanks** slide.

You are allowed to:

- Modify this template.
- Use it for both personal and commercial purposes.
- Hide or delete the “Thanks” slide and the mention to Slidesgo in the credits.
- Share this template in an editable format with people who are not part of your team.

You are not allowed to:

- Sublicense, sell or rent this Slidesgo Template (or a modified version of this Slidesgo Template).
- Distribute this Slidesgo Template (or a modified version of this Slidesgo Template) or include it in a database or in any other product or service that offers downloadable images, icons or presentations that may be subject to distribution or resale.
- Use any of the elements that are part of this Slidesgo Template in an isolated and separated way from this Template.
- Register any of the elements that are part of this template as a trademark or logo, or register it as a work in an intellectual property registry or similar.

For more information about editing slides, please read our FAQs or visit our blog:
https://slidesgo.com/faqs and **https://slidesgo.com/slidesgo-school**

Fonts & colors used

This presentation has been made using the following fonts:

Source Code Pro
(https://fonts.google.com/specimen/Source+Code+Pro)

Storyset

Create your Story with our illustrated concepts. Choose the style you like the most, edit its colors, pick the background and layers you want to show and bring them to life with the animator panel! It will boost your presentation. Check out **how it works**.

Pana

Amico

Bro

Rafiki

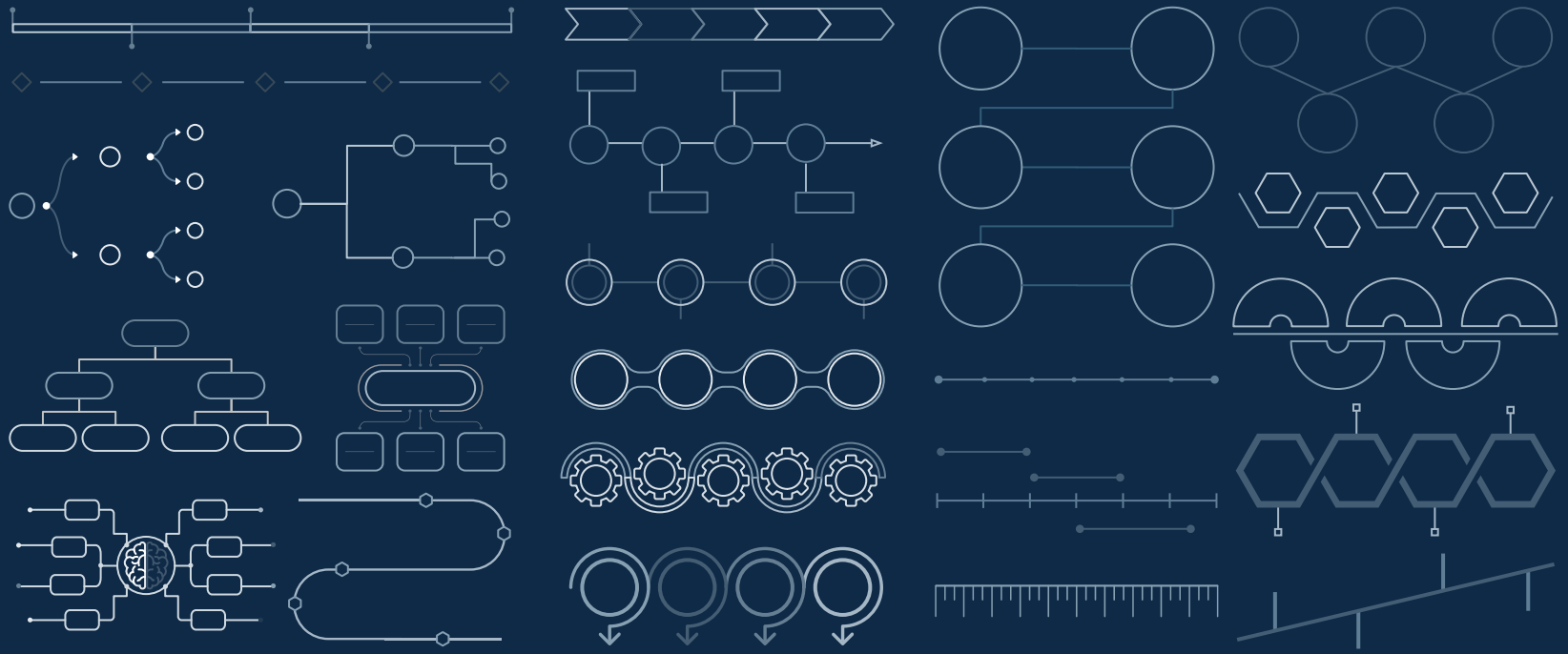
Cuate

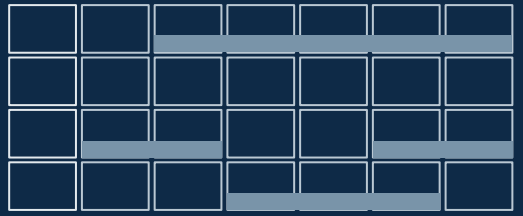
Use our editable graphic resources...

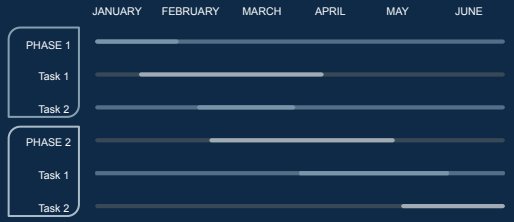
You can easily **resize** these resources without losing quality. To **change the color**, just ungroup the resource and click on the object you want to change. Then, click on the paint bucket and select the color you want. Group the resource again when you're done. You can also look for more **infographics** on Slidesgo.

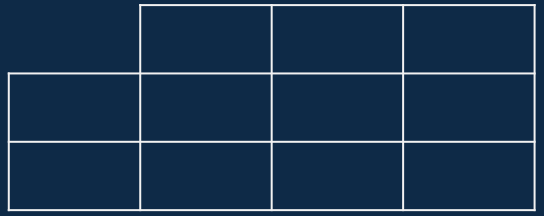


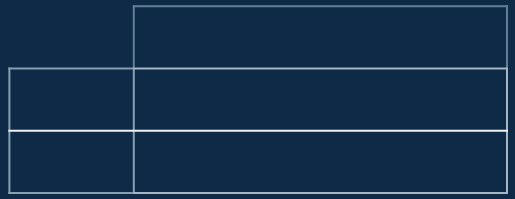


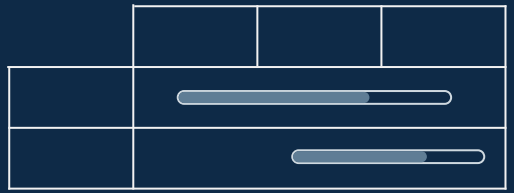



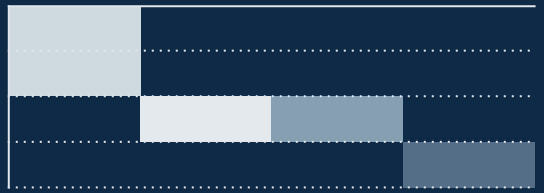


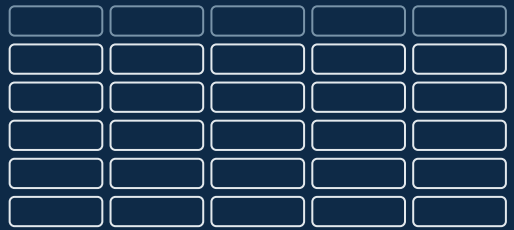

JANUARY
FEBRUARY
MARCH
APRIL
MAY
JUNE
PHASE 1
Task 1
Task 2
PHASE 2
Task 1
Task 2

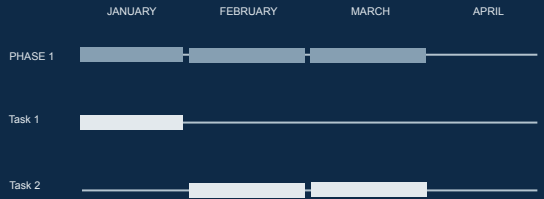


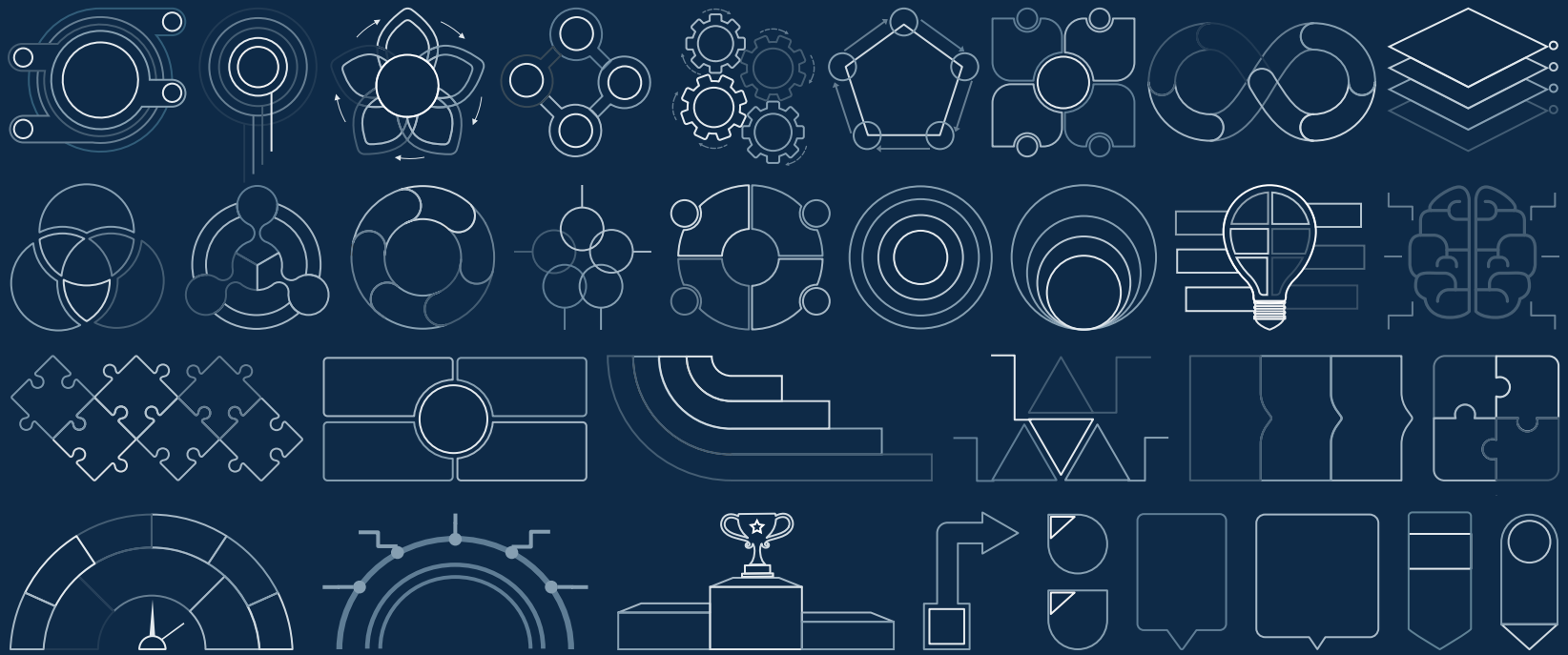


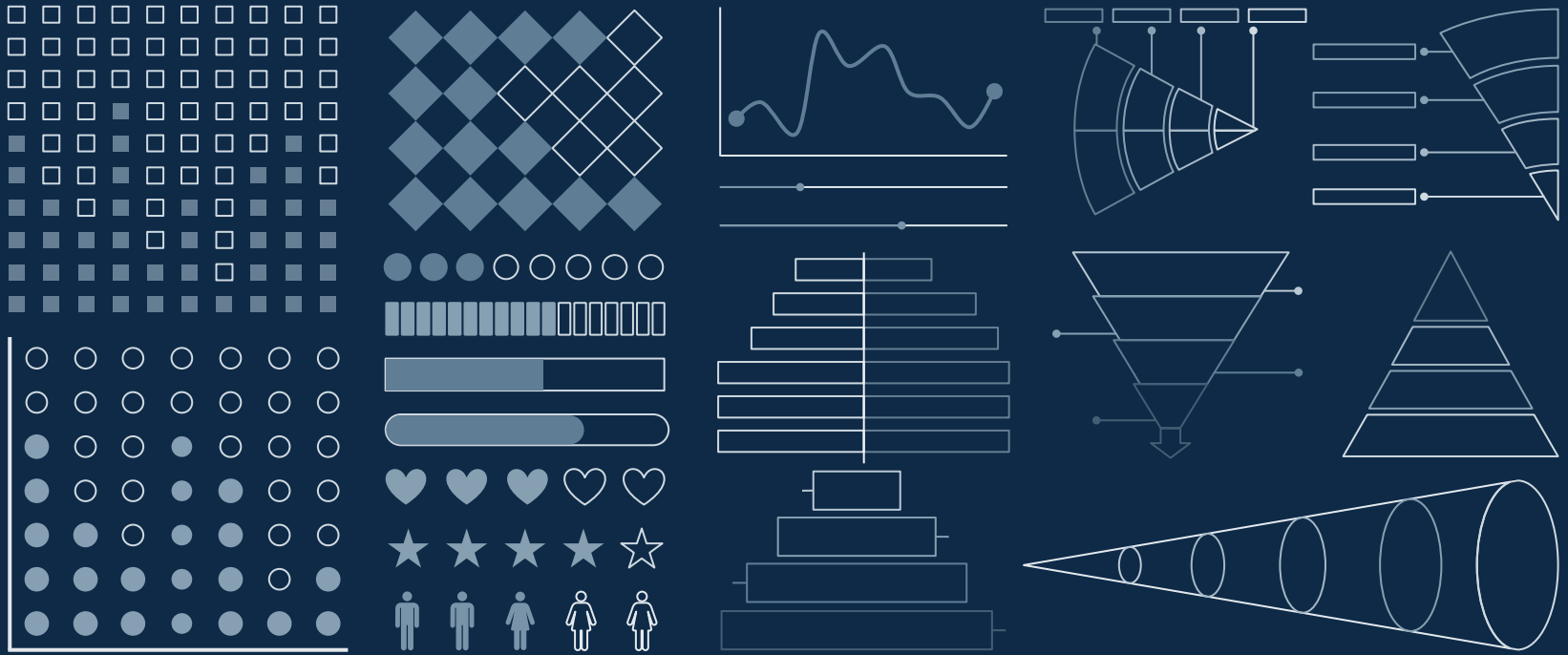







JANUARY
FEBRUARY
MARCH
APRIL
PHASE 1
Task 1
Task 2





...and our sets of editable icons

You can **resize** these icons without losing quality.
You can **change the stroke and fill color**; just select the icon and click on the **paint bucket/pen**.
In Google Slides, you can also use **Flaticon's extension**, allowing you to customize and add even more icons.

Educational Icons

Medical Icons

Business Icons

Teamwork Icons

Help & Support Icons

Avatar Icons

Creative Process Icons

Performing Arts Icons

Nature Icons

SEO & Marketing Icons


slidesgo