Top down approach

-And how to apply-

Journey to your best – Challenge 0

Table of contents

01

Top down approach

Definition of Top down approach

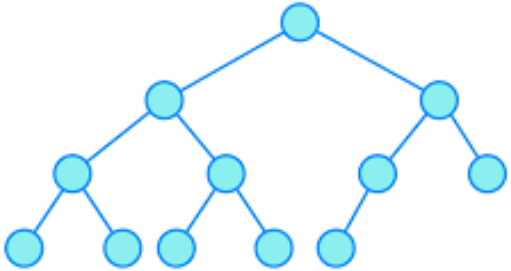
The top-down approach is a methodology that initiates with a comprehensive overview and systematically progresses towards more intricate details.

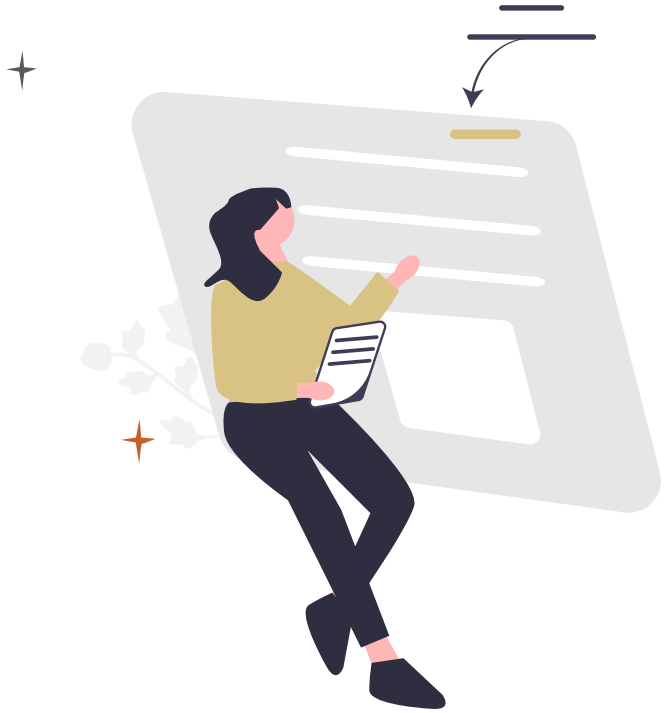
The top-down approach is a methodology or strategy used in various fields, including software development, management, and problem-solving.

Features of Top down approach

Hierarchy: It's like building from the top-down. Top-level decision-makers start with a broad goal and work backward to plan the actions needed for different groups and individuals to reach it

Sequential: Progression occurs from the most abstract or general level to the most detailed level.





02

Plus and Cons

Advantage of Top down approach

Clarity

Provides a clear overall picture before delving into specifics.

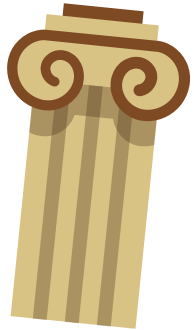
Structured

Offers a structured and organized approach to problem-solving.

Efficiency

Can lead to more efficient solutions by tackling the most critical issues first.

Disadvantages of Top down approach



Limited Flexibility

May not adapt well to rapidly changing environments.

Hard to change

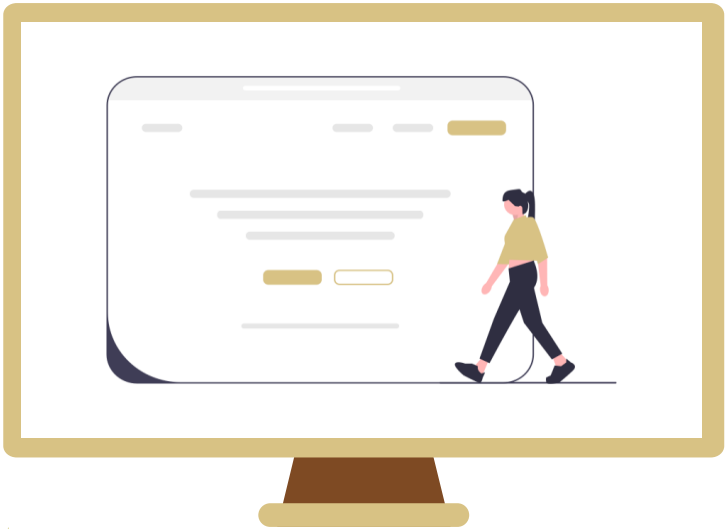
Once the initial design or plan is established, there might be reluctance to change.

03 Top down approach in real life

In the tech world, the top-down approach is like a cool strategy used in different areas of development and management.

In business and management

In business and management, it involves setting overall goals and strategies at the highest levels, which are then broken down into specific objectives for each department or team.



In the tech industry

Programming

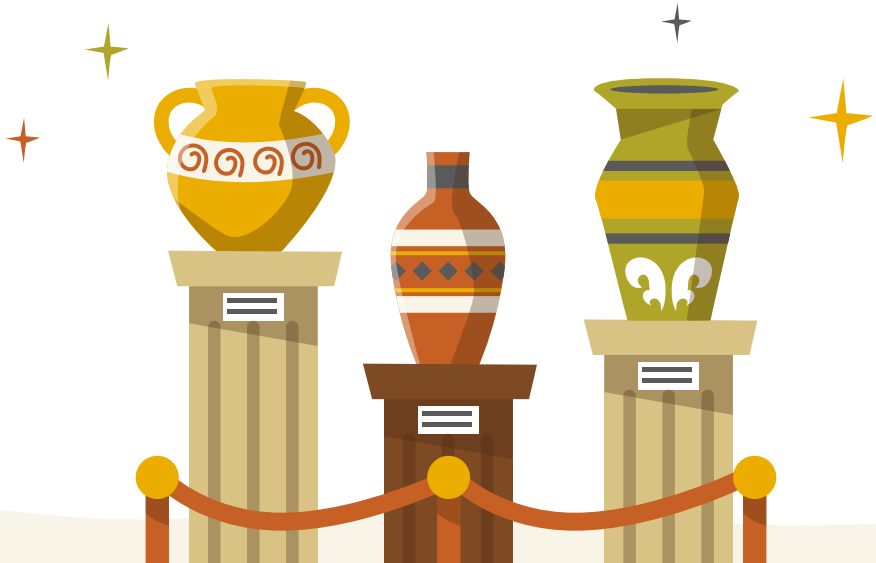
Writing code by first defining high-level modules and functions before diving into the implementation details.

Design

Designing a technology system by starting with the overall architecture and progressively detailing components.

04 How to apply

How to use this approach



How to apply

Step 1

Define the overall goals and objectives of the system.

Step 2

Break down the system into smaller components or modules.

Step 3

Develop each component individually.

Step 4

Test and debug each component

How to apply

Step 5

Integrate the components into the larger system

Step 6

Test the entire system to ensure that it is functioning.

Step 7

Maintain and update the system casually.

Thanks!