IIT Madras
BSc Degree

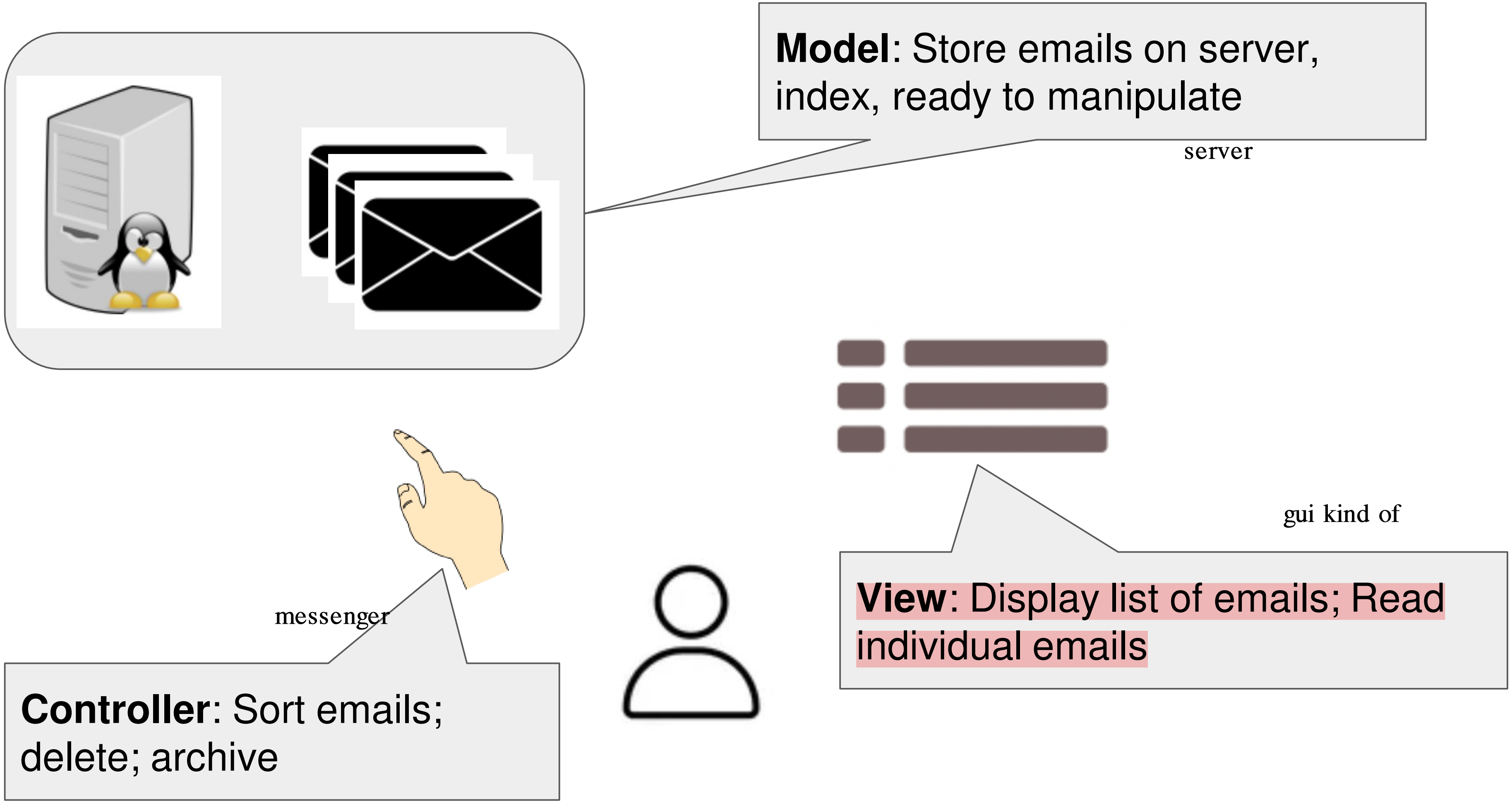
Views

Outline

- MVC paradigm
- Views and User Interfaces
- Tools and Techniques
- Accessibility

MVC

- Model
- **View**
- Controller



Model: Store emails on server, index, ready to manipulate

server

View: Display list of emails; Read individual emails

gui kind of

Controller: Sort emails; delete; archive

messenger

Model-View-Controller

- Origins: Smalltalk-80
- Separation of responsibilities - **Abstraction**
- Roots in Object-Oriented GUI development

Model-View-Controller

- Origins: Smalltalk-80
- Separation of responsibilities - **Abstraction**
- Roots in Object-Oriented GUI development

Design patterns

- Common software patterns
- **Model**: Application object
- **View**: Screen representation
- **Controller**: How user *interface* reacts to user *input*

Running Example

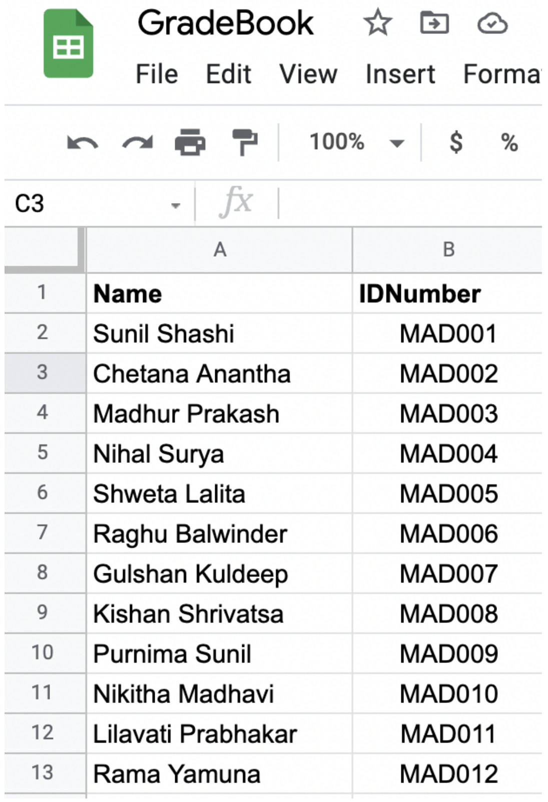
Student Gradebook

- Input data: for **Model**
  - Student list
  - Course list
  - Student-Course marks

Running Example

Student Gradebook

- Input data: for **Model**
  - Student list
  - Course list
  - Student-Course marks

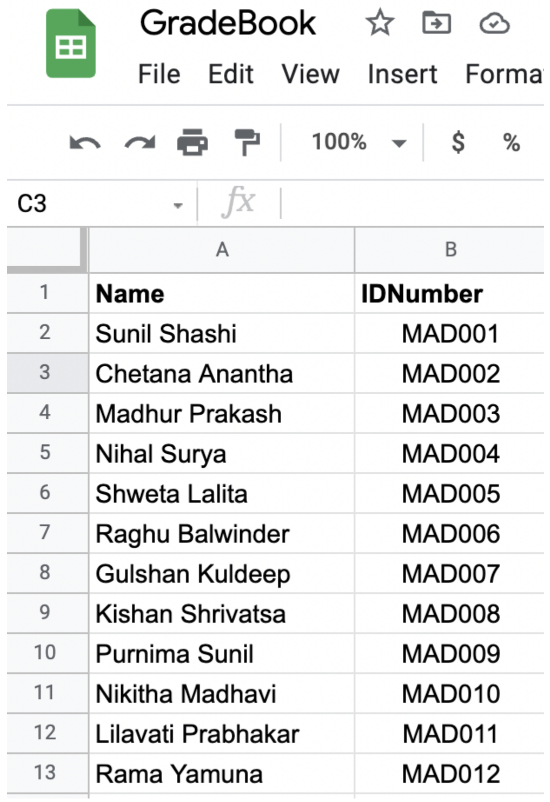


	A	B
1	**Name**	**IDNumber**
2	Sunil Shashi	MAD001
3	Chetana Anantha	MAD002
4	Madhur Prakash	MAD003
5	Nihal Surya	MAD004
6	Shweta Lalita	MAD005
7	Raghu Balwinder	MAD006
8	Gulshan Kuldeep	MAD007
9	Kishan Shrivatsa	MAD008
10	Purnima Sunil	MAD009
11	Nikitha Madhavi	MAD010
12	Lilavati Prabhakar	MAD011
13	Rama Yamuna	MAD012

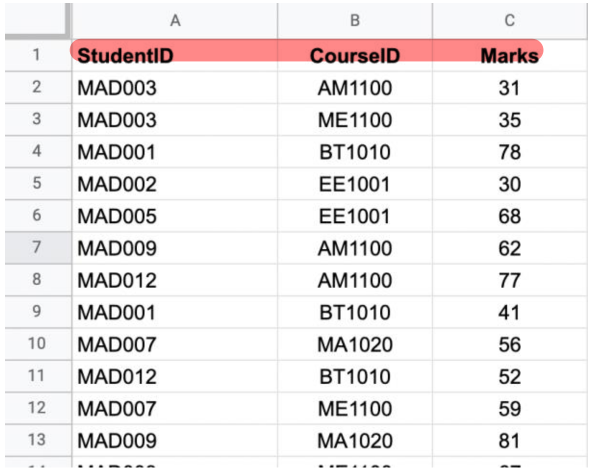
Running Example

Student Gradebook

- Input data: for **Model**
    - Student list
    - Course list
    - Student-Course marks



	A	B
1	**Name**	**IDNumber**
2	Sunil Shashi	MAD001
3	Chetana Anantha	MAD002
4	Madhur Prakash	MAD003
5	Nihal Surya	MAD004
6	Shweta Lalita	MAD005
7	Raghu Balwinder	MAD006
8	Gulshan Kuldeep	MAD007
9	Kishan Shrivatsa	MAD008
10	Purnima Sunil	MAD009
11	Nikitha Madhavi	MAD010
12	Lilavati Prabhakar	MAD011
13	Rama Yamuna	MAD012



	A	B	C
1	**StudentID**	**CourseID**	**Marks**
2	MAD003	AM1100	31
3	MAD003	ME1100	35
4	MAD001	BT1010	78
5	MAD002	EE1001	30
6	MAD005	EE1001	68
7	MAD009	AM1100	62
8	MAD012	AM1100	77
9	MAD001	BT1010	41
10	MAD007	MA1020	56
11	MAD012	BT1010	52
12	MAD007	ME1100	59
13	MAD009	MA1020	81

Running Example

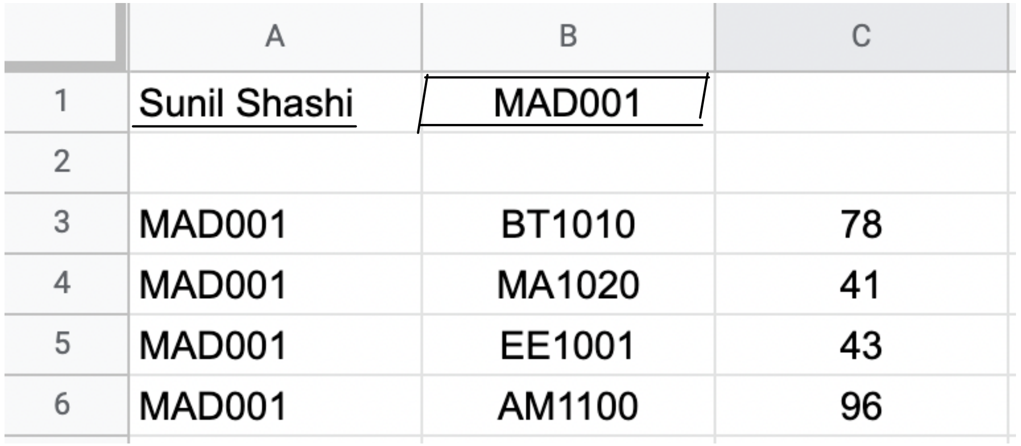
Student Gradebook

- Outputs: for **Views**

Running Example

Student Gradebook

- Outputs: for **Views**
  - marks for individual student

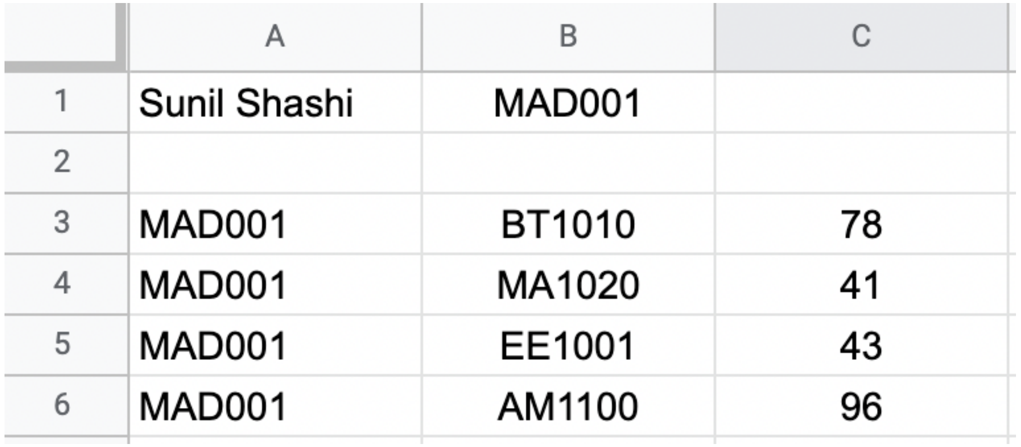


	A	B	C
1	Sunil Shashi	MAD001	
2			
3	MAD001	BT1010	78
4	MAD001	MA1020	41
5	MAD001	EE1001	43
6	MAD001	AM1100	96

Running Example

Student Gradebook

- Outputs: for **Views**
  - marks for individual student
  - summary for course
  - histograms



	A	B	C
1	Sunil Shashi	MAD001	
2			
3	MAD001	BT1010	78
4	MAD001	MA1020	41
5	MAD001	EE1001	43
6	MAD001	AM1100	96

Running Example

Student Gradebook

- Modifications: for **Controllers**
    - add new students
    - add new courses
    - modify marks in course

Views

User Interface Design

View

User Interface

- Screen
- Audio
- Vibration (haptic)
- Motor (door open/close)

View

User Interface

- Screen
- Audio
- Vibration (haptic)
- Motor (door open/close)

User Interaction

- Keyboard / Mouse
- Touchscreen
- Spoken voice
- Custom buttons

User Interaction

- Determined by hardware constraints
- Different target devices possible
- User-Agent information useful to identify context

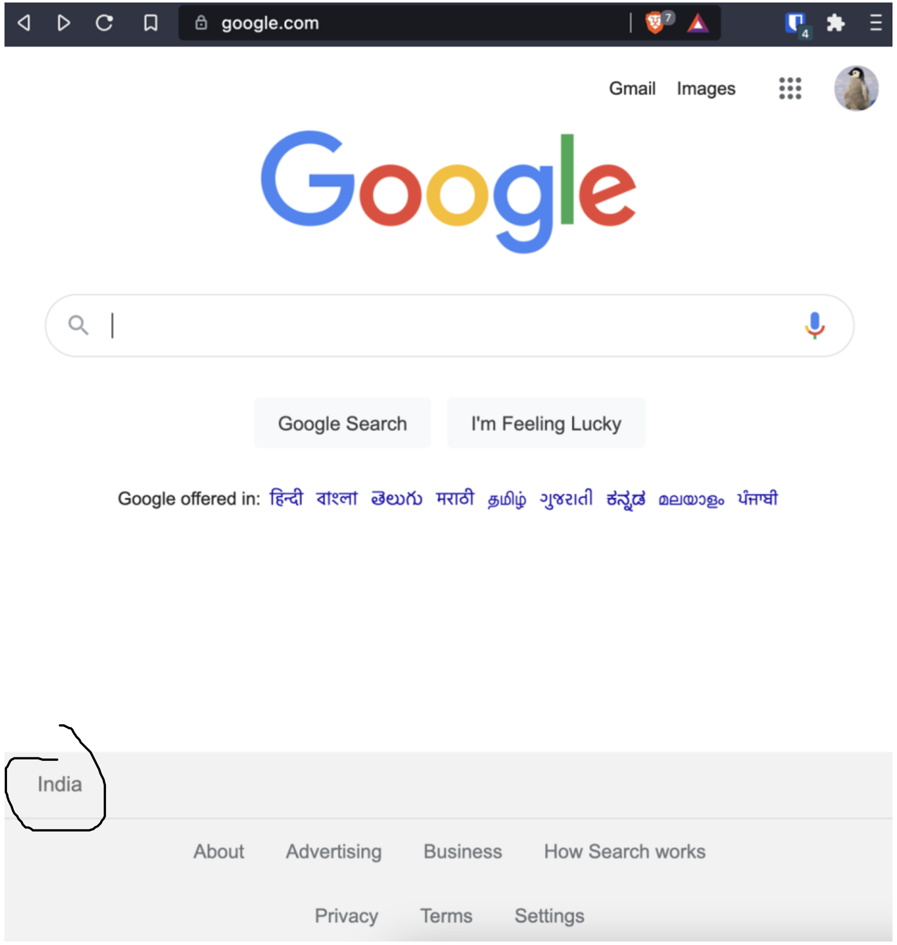
User Interaction

- Determined by hardware constraints
- Different target devices possible
- User-Agent information useful to identify context

May not be under designer control

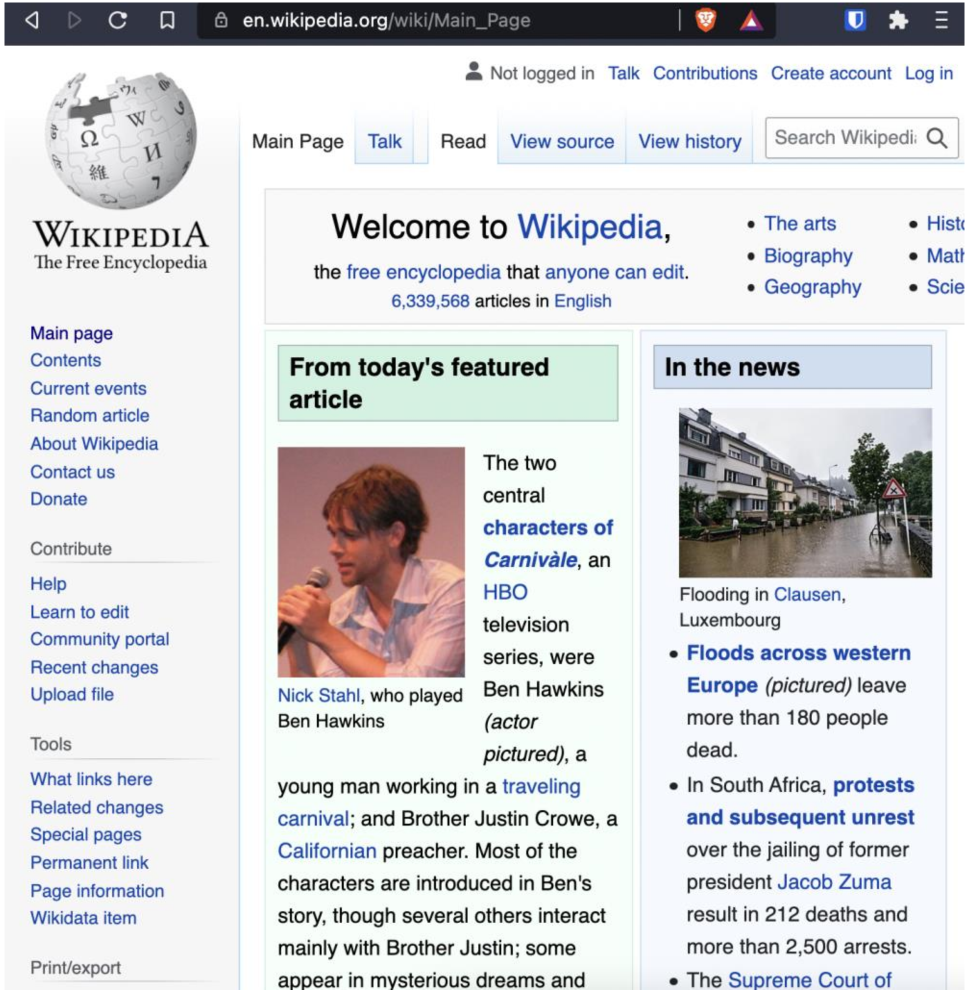
Types of Views

- Fully static



Types of Views

- Fully static
- ~~Partly dynamic~~



en.wikipedia.org/wiki/Main_Page

Not logged in Talk Contributions Create account Log in

Main Page Talk Read View source View history Search Wikipedi

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Wikidata item

Print/export

Welcome to Wikipedia,
the free encyclopedia that anyone can edit.
6,339,568 articles in English

- The arts
- Biography
- Geography
- Hist
- Math
- Scie

From today's featured article

Nick Stahl, who played Ben Hawkins

The two central characters of *Carnivàle*, an HBO television series, were Ben Hawkins *(actor pictured)*, a young man working in a traveling carnival; and Brother Justin Crowe, a Californian preacher. Most of the characters are introduced in Ben's story, though several others interact mainly with Brother Justin; some appear in mysterious dreams and

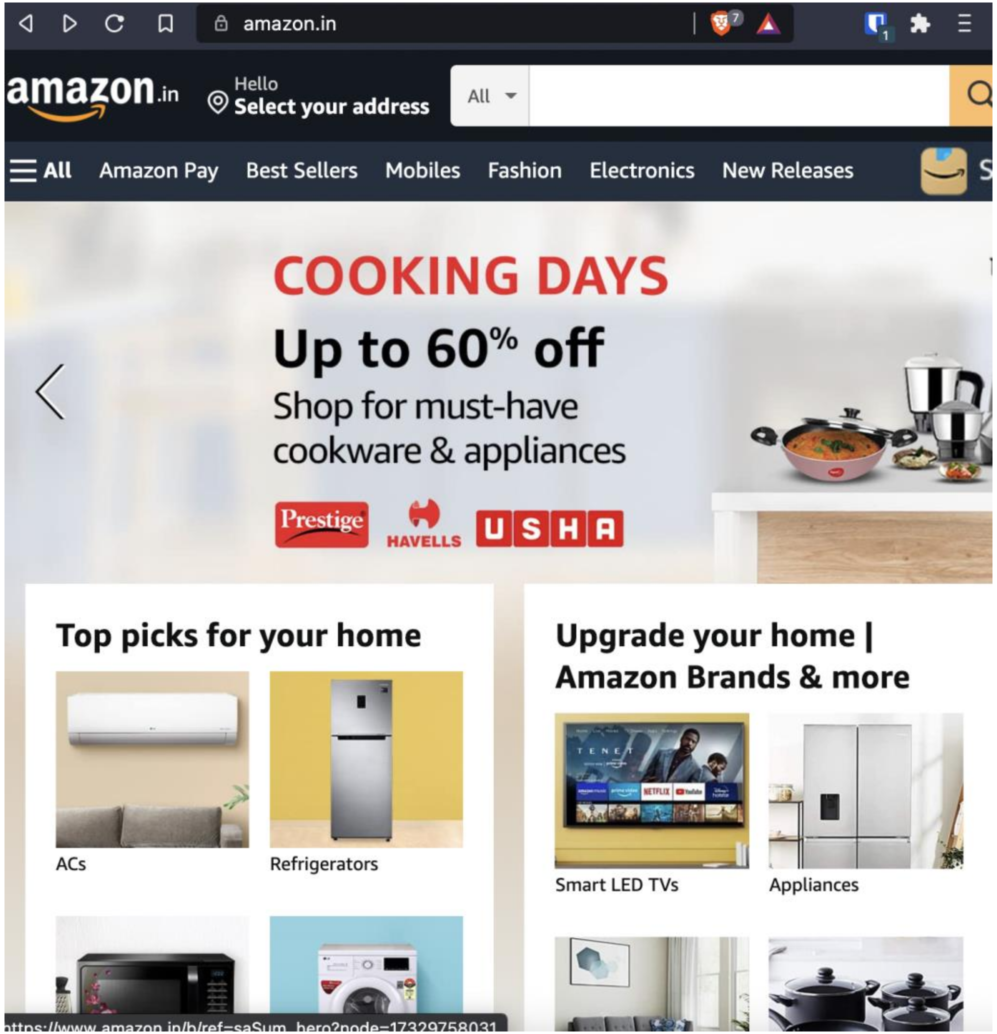
In the news

Flooding in Clausen, Luxembourg

- **Floods across western Europe** *(pictured)* leave more than 180 people dead.
- In South Africa, **protests and subsequent unrest** over the jailing of former president Jacob Zuma result in 212 deaths and more than 2,500 arrests.
- The Supreme Court of

Types of Views

- Fully static
- Partly dynamic
- Mostly dynamic



Output

- HTML - most commonly used - direct rendering
- Dynamic images
- JSON / XML - machine readable

Output

- HTML - most commonly used - direct rendering
- Dynamic images
- JSON / XML - machine readable

View - any “representation” useful to another entity

User Interface Design

- Design for interaction with user
- Goals:
    - **Simple** - easy for user to understand and use
    - **Efficient** - user achieves goal with minimal effort

User Interface Design

- Design for interaction with user
- Goals:
  - **Simple** - easy for user to understand and use
  - **Efficient** - user achieves goal with minimal effort
- Aesthetics

User Interface Design

- Design for interaction with user
- Goals:
  - **Simple** - easy for user to understand and use
  - **Efficient** - user achieves goal with minimal effort
- Aesthetics
- Accessibility

Systematic Process

- Functionality requirements gathering - what is needed?

Systematic Process

- Functionality requirements gathering - what is needed?
- User and Task analysis - user preferences, task needs

Systematic Process

- Functionality requirements gathering - what is needed?
- User and Task analysis - user preferences, task needs
- Prototyping - wireframes, mockups

Systematic Process

- Functionality requirements gathering - what is needed?
- User and Task analysis - user preferences, task needs
- Prototyping - wireframes, mockups
- Testing - user acceptance, usability, accessibility

Guidelines / Heuristics

Jakob Nielsen’s heuristics for design

https://www.nngroup.com/articles/ten-usability-heuristics/

- Not specific to web apps, or even software UI design
- Very useful and relevant

General principles

- Consistency
- Simple and minimal steps
- Simple language
- Minimal and aesthetically pleasing

Tools

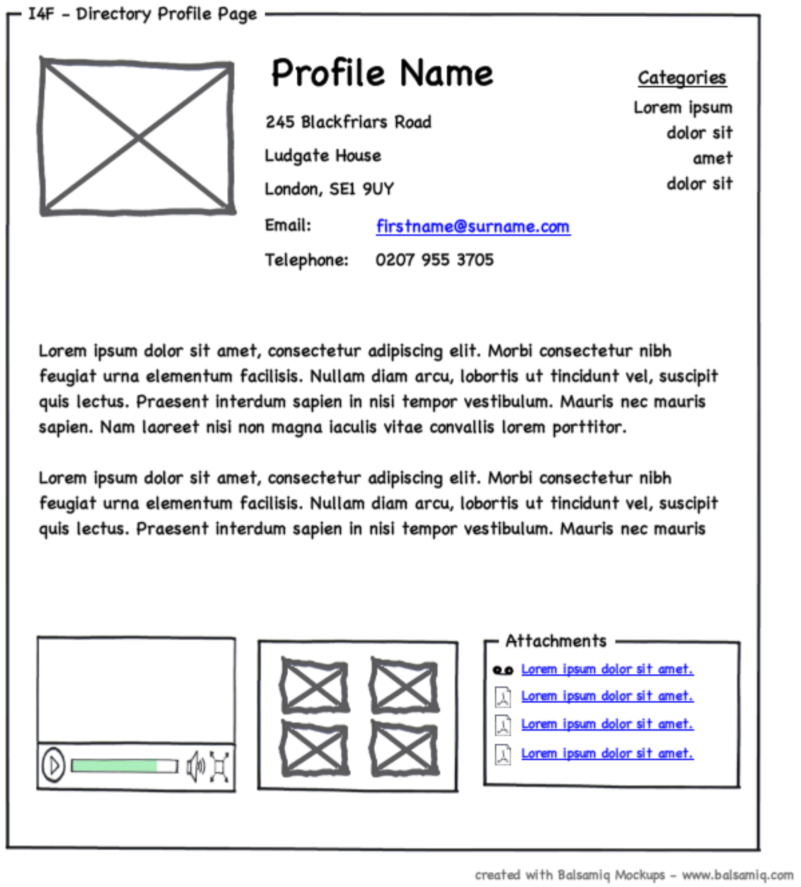
- Wireframes
- HTML generation
- Templates

Wireframes

- Visual guide to represent **structure** of web page
- Information design
- Navigation design
- User interface design

Wireframes

- Visual guide to represent **structure** of web page
- Information design
- Navigation design
- User interface design


I4F - Directory Profile Page
Profile Name
Categories
Lorem ipsum
dolor sit
amet
dolor sit
245 Blackfriars Road
Ludgate House
London, SE1 9UY
Email: firstname@surname.com
Telephone: 0207 955 3705
Attachments
Lorem ipsum dolor sit amet.
Lorem ipsum dolor sit amet.
Lorem ipsum dolor sit amet.
Lorem ipsum dolor sit amet.
created with Balsamiq Mockups - www.balsamiq.com

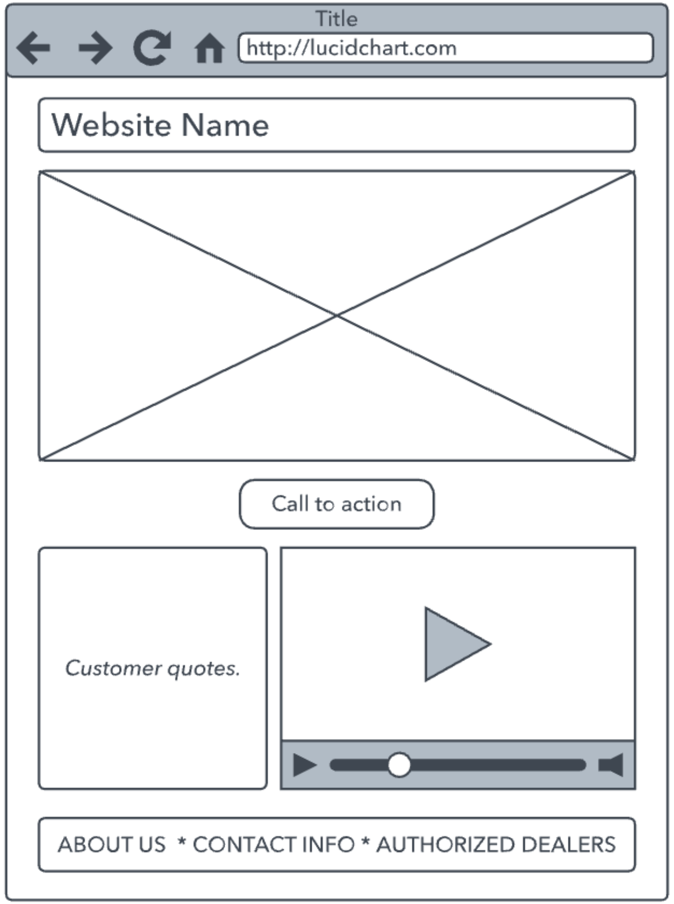
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi consectetur nibh feugiat urna elementum facilisis. Nullam diam arcu, lobortis ut tincidunt vel, suscipit quis lectus. Praesent interdum sapien in nisi tempor vestibulum. Mauris nec mauris sapien. Nam laoreet nisi non magna iaculis vitae convallis lorem porttitor.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi consectetur nibh feugiat urna elementum facilisis. Nullam diam arcu, lobortis ut tincidunt vel, suscipit quis lectus. Praesent interdum sapien in nisi tempor vestibulum. Mauris nec mauris

Src: https://en.wikipedia.org/wiki/Website_wireframe

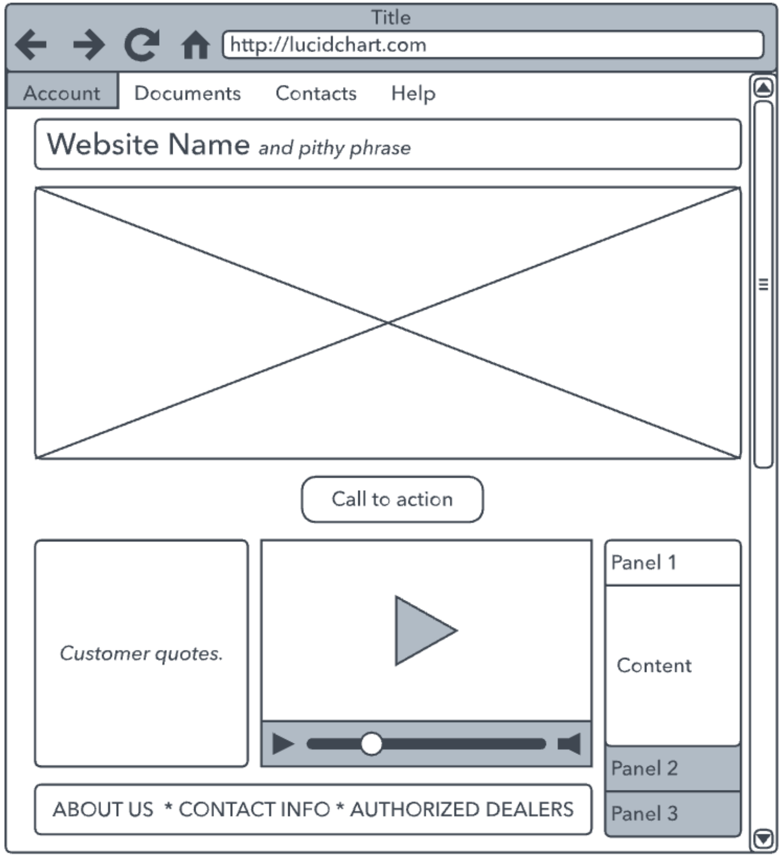
Example Tools: LucidChart

https://www.lucidchart.com/pages/wireframe



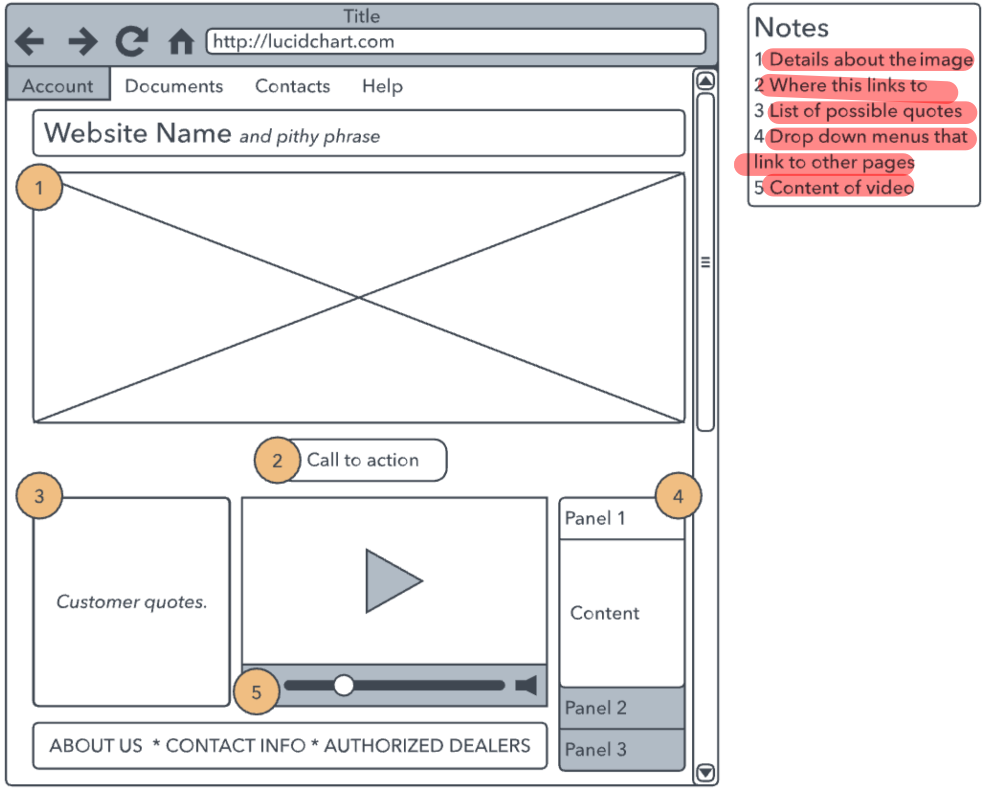
Example Tools: LucidChart

https://www.lucidchart.com/pages/wireframe



Example Tools: LucidChart

https://www.lucidchart.com/pages/wireframe



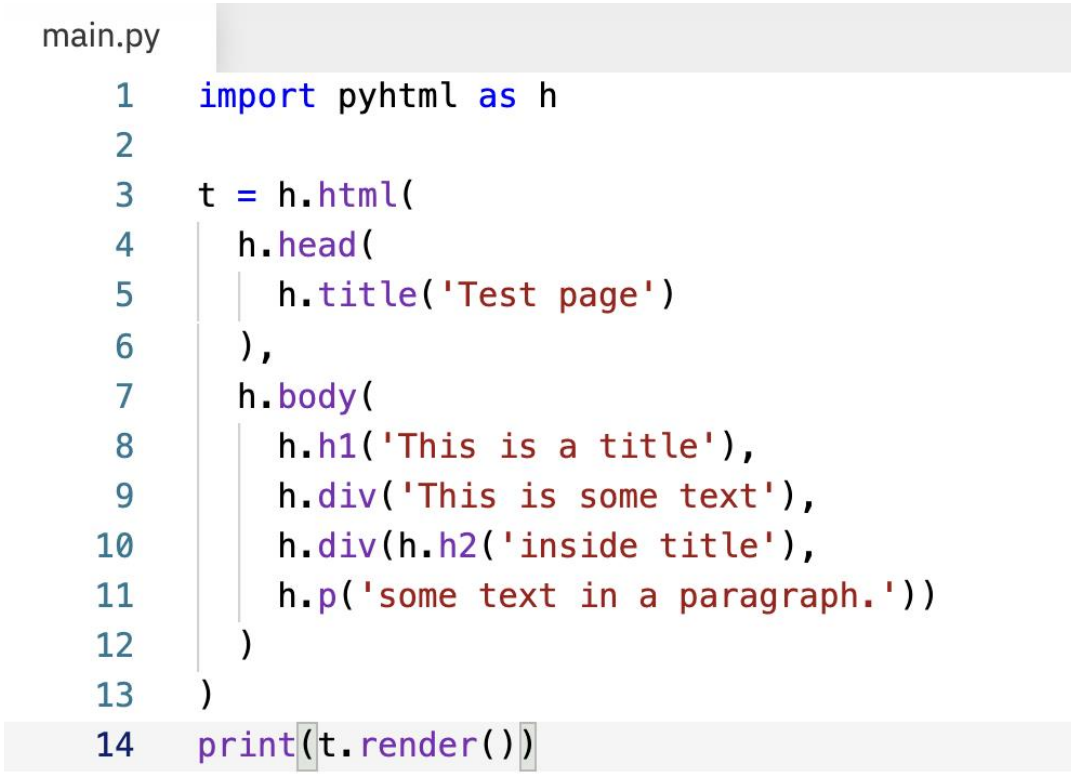
Title
http://lucidchart.com
Account
Documents
Contacts
Help
Website Name *and pithy phrase*
1
2
Call to action
3
Customer quotes.
4
Panel 1
Content
Panel 2
Panel 3
5
ABOUT US * CONTACT INFO * AUTHORIZED DEALERS
Notes
1 Details about the image
2 Where this links to
3 List of possible quotes
4 Drop down menus that link to other pages
5 Content of video

Programmatic HTML generation: PyHTML

- Composable functions - each function generates a specific output
- Example:
    - to generate a h1 heading: function should return

      “<h1>text of heading</h1>”

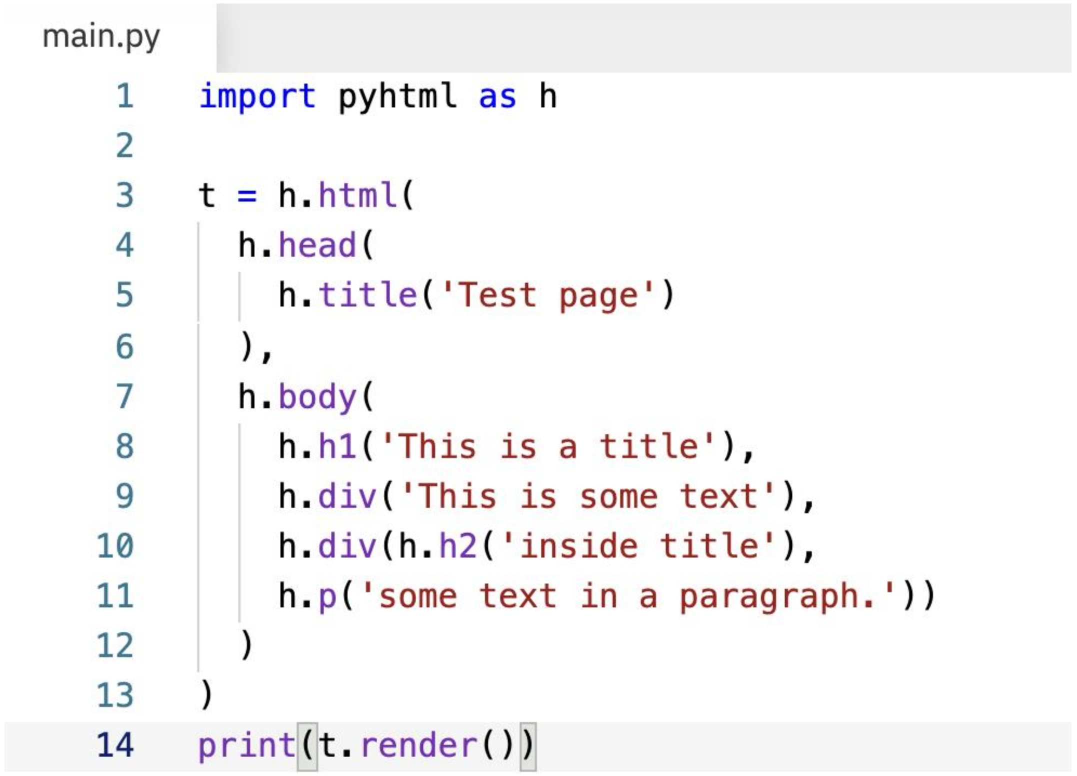
Programmatic HTML generation: PyHTML



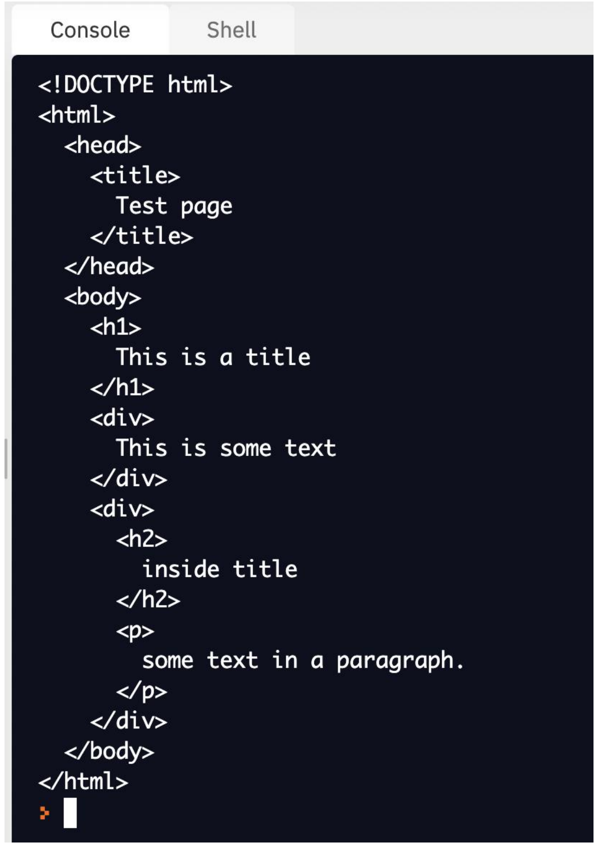
main.py

```python
import pyhtml as h

t = h.html(
    h.head(
        h.title('Test page')
    ),
    h.body(
        h.h1('This is a title'),
        h.div('This is some text'),
        h.div(h.h2('inside title'),
        h.p('some text in a paragraph.'))
    )
)
print(t.render())
```

Programmatic HTML generation: PyHTML



```python
import pyhtml as h

t = h.html(
    h.head(
        h.title('Test page')
    ),
    h.body(
        h.h1('This is a title'),
        h.div('This is some text'),
        h.div(h.h2('inside title'),
        h.p('some text in a paragraph.'))
    )
)
print(t.render())
```



```
<!DOCTYPE html>
<html>
  <head>
    <title>
      Test page
    </title>
  </head>
  <body>
    <h1>
      This is a title
    </h1>
    <div>
      This is some text
    </div>
    <div>
      <h2>
        inside title
      </h2>
      <p>
        some text in a paragraph.
      </p>
    </div>
  </body>
</html>
›
```

More complex HTML

```
def f_table(ctx):
    return (tr(
        td(cell) for cell in row
    ) for row in ctx['table'])
```

Templates

- Standard template text
- Placeholders / Variables
- Basic (very limited) programmability
- Examples:
    - Python inbuilt String Templates - good for simple tasks
    - Jinja2 - used by Flask
    - Genshi
    - Mako
    - … - *just pick one and go with it*

Jinja

- Ties in closely with Flask
- Template functionality with detailed API

Remember: templates can generate any output, not just HTML

Accessibility

Accessibility

- Various forms of disability or impairment
    - Vision
    - Speech
    - Touch
    - Sensor-Motor
- Can a page be accessed by people with impairments?
- How can the accessibility of a page be improved?

W3.org - World Wide Web Consortium (W3C) - accessibility guidelines

https://www.w3.org/WAI/fundamentals/accessibility-principles/

Standards

Interplay between many components of a page:

- Web content: HTML, images, scripts etc.
- User-agents: desktop browser, mobile browser, speech-oriented browser, assistive devices
- Authoring tools: text editor, word processor, compiler

Principle - Perceivable

- Provide **text alternatives** for non-text content.
- Provide **captions and other alternatives** for multimedia.
- Create content that can be **presented in different ways**, including by assistive technologies, without losing meaning.
- Make it easier for users to **see and hear content**.

Principle - Operable

- Make all functionality available from a keyboard.
- Give users enough time to read and use content.
- Do not use content that causes seizures or physical reactions.
- Help users navigate and find content.
- Make it easier to use inputs other than keyboard.

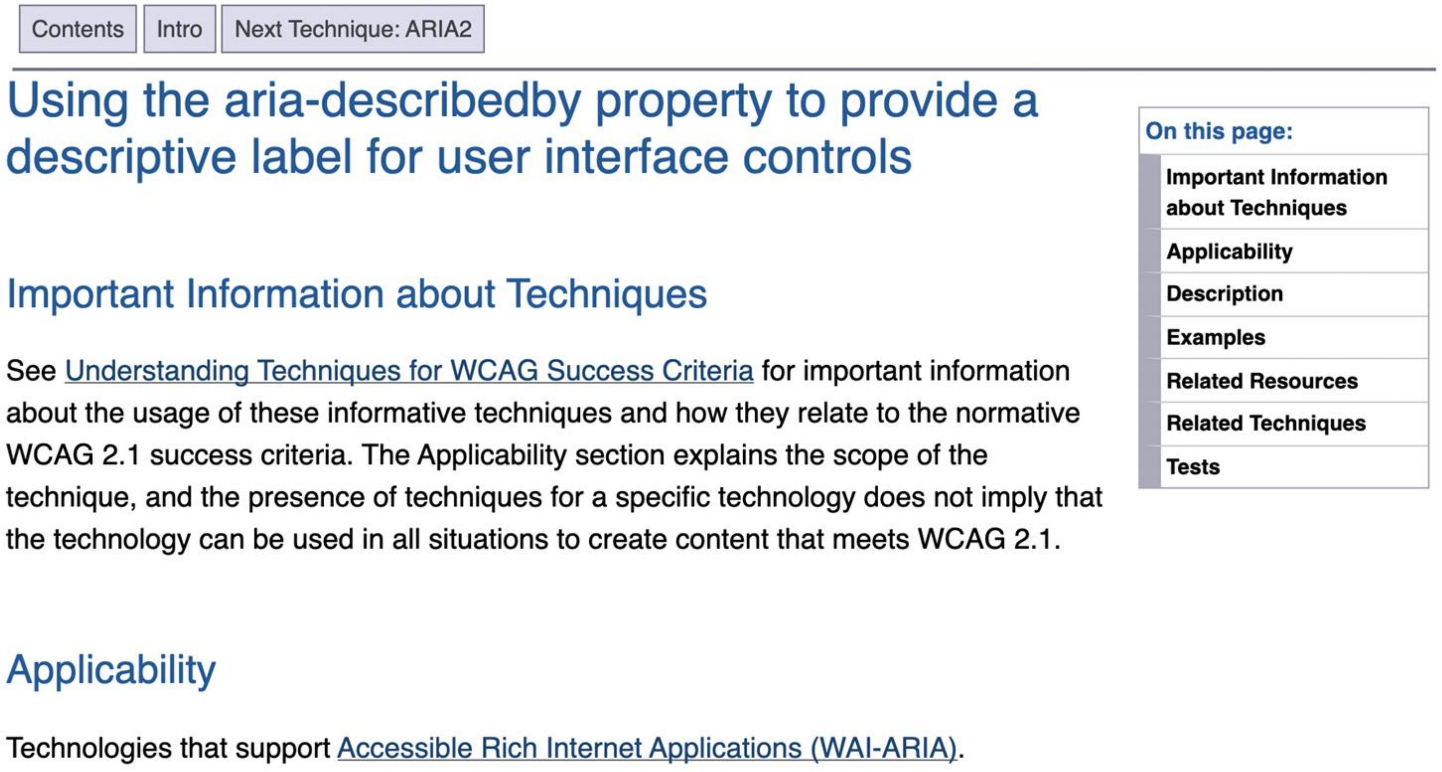
Principle - Understandable

- Make text readable and understandable.
- Make content appear and operate in predictable ways.
- Help users avoid and correct mistakes.

Principle - Robust

- Maximize compatibility with current and future user tools.

Techniques



Using the aria-describedby property to provide a descriptive label for user interface controls

On this page:

Important Information about Techniques

See Understanding Techniques for WCAG Success Criteria for important information about the usage of these informative techniques and how they relate to the normative WCAG 2.1 success criteria. The Applicability section explains the scope of the technique, and the presence of techniques for a specific technology does not imply that the technology can be used in all situations to create content that meets WCAG 2.1.

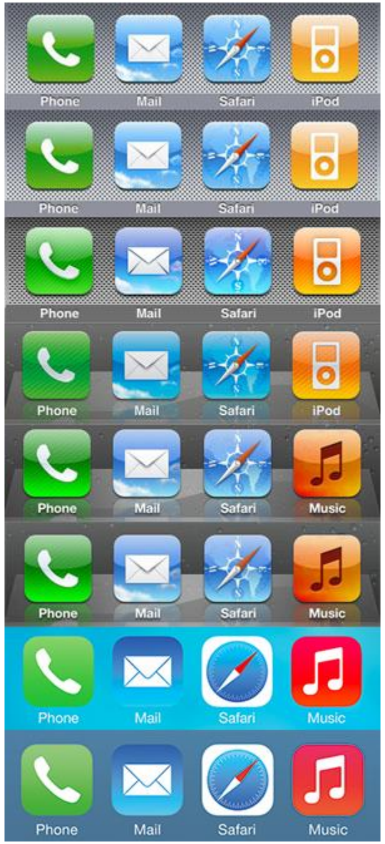
Applicability

Technologies that support Accessible Rich Internet Applications (WAI-ARIA).

Aesthetics

- Visual appearance
- VERY important
- Simplicity preferred

Can vary with time!



From: The Verge: iOS: A Visual History

Summary

- View - any output seen by human or machine
- User-interface and User-interaction guidelines
- Accessibility is a core concept!
- Tools for automatic generation, consistent layout