# WEB TECHNOLOGIES

# How the Web works

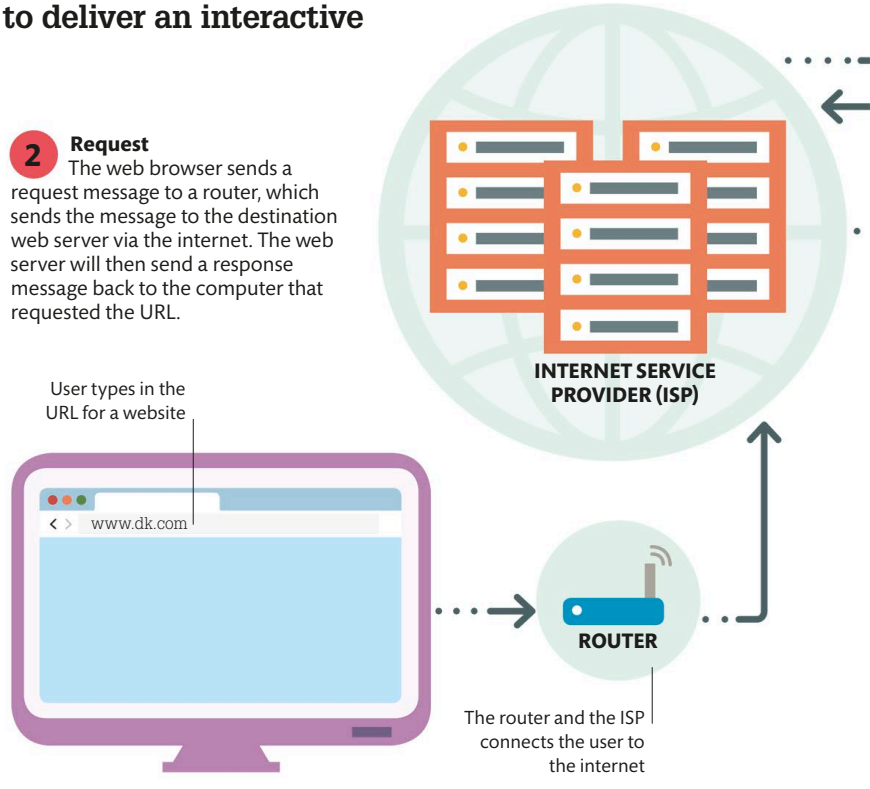
**The World Wide Web is a set of technologies that work together to allow information to be shared between computers via the internet. The Web is characterized by its combination of text, images, video, and audio to deliver an interactive multimedia experience.**

## Connecting to a website

The Web is based on a client/server model. A browser is a client that requests a web page from a server. The server then responds to the request by sending an HTML file. The content of each request is determined by the communication protocol being used. Hypertext Transfer Protocol (HTTP) is the most common protocol used over the internet—a global network created from connections between billions of devices.

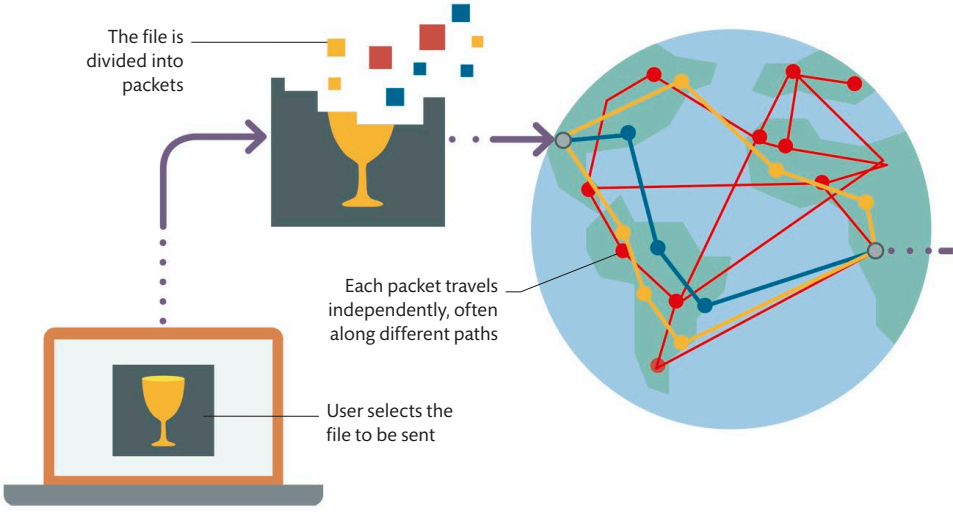
**1 Enter web page URL**
The process begins when a user enters a Uniform Resource Locator (URL) into the address bar of an internet browser. This URL contains the address of the requested web page and can be used to locate the web server that hosts the website.

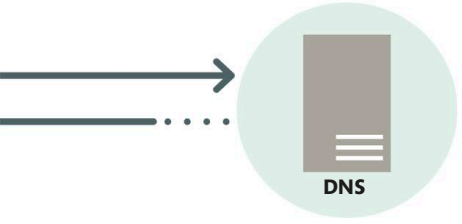
**2 Request**
The web browser sends a request message to a router, which sends the message to the destination web server via the internet. The web server will then send a response message back to the computer that requested the URL.




## Packets and IP routing

All communication over the Web is done by dividing the request into smaller segments of data called packets. These packets are routed from the source to the destination, where they are reassembled into the original message. The networks that convey data in packets are called "packet switched networks." Packets consist of two parts: information and data. Information defines where and how to send the data, while data is the content that the packet is trying to deliver.

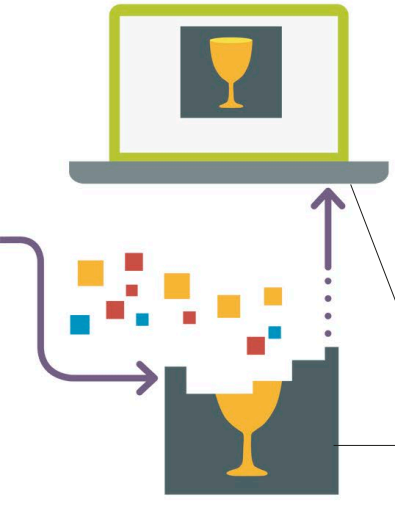





**3 Finding the website**
The Domain Name Systems (DNS) protocol allows the browser to convert the human-friendly text into an IP address. This address is then used by the routers to find a path to the web server. The request may be passed along many routers before it arrives at the destination web server.

The website is displayed on the user's hardware


**4 View the page**
The web server receives the request and returns an HTML file as a response to the browser. The browser reads the contents of the HTML document and renders the text, images, and data onscreen.



**Why packets?**
To allow communication channels to be used more efficiently, images, text, and even basic HTTP requests are broken down and transferred piece by piece. Each piece of data has a packet sequence, which tells the receiving server how to reassemble the information.

The reassembled file can be viewed by the receiver

The file is reassembled in the right order

## Protocols

A protocol is a set of rules that governs the communication between two entities. Protocols on the Web exist to manage the communication between the client browser and the web server. Network protocols are structured as a series of layers. Each layer is designed for a specific purpose and exists on both the sending and receiving hosts.

**Application layer protocol**
Defines how an application must format its data so that it can communicate with other applications. For example, HTTP and File Transfer Protocol (FTP) define how a web browser can communicate with a web server.

**Link layer protocol**
Defines how data can be sent from one network to another by using routers to find the destination computer and deliver the message.

**Transport layer protocol**
Defines how to manage communications by maintaining sessions between the source and destination computers and combining the received packets back into the correct order.

**Web protocols**
Transmission Control Protocol (TCP) manages the sessions and ordering of the packets received by the browser. Internet Protocol (IP) handles routing of data between the client and the server. HTTP/FTP/UDP (User Datagram Protocol) defines the messages being sent between the browser and the server.

### HTTP

HTTP is an application level protocol that describes how a client can format and send a request message to a server and how the server can format and reply with a response message.

- The GET method retrieves data.
- The POST method updates data.
- The PUT method creates data.
- The DELETE method removes data.

# Code editors

**One of the most important tools for programmers, code editors are specifically designed for editing the source code of computer programs. They can be stand-alone applications or part of any IDE (see p.23) or web browser. A number of code editors are available online, all customized to fit specific work situations or programming languages.**

## Code editor tools

Simple text editors, such as Notepad, can be used to write code, but they cannot enhance or ease the process of code editing. The code editors available online have specialized functionalities, or certain built-in features, that simplify and accelerate the process of editing. These elements automate common repetitive tasks and assist the programmer to write better software by identifying problems and debugging code. Some of the most useful code editor tools are given here.

**Syntax highlighting**
Displays different parts of the code in different colors, making the code easier to read. For example, HTML tags are highlighted in one color and comments are highlighted in another color.

**Printing**
Enables the programmer to print a hard copy of the code. The output can then be shared and used as a tool to facilitate communication and problem solving.

**Multiview**
Allows the programmer to view multiple files side by side. Some code editors even allow two instances of the same file to be viewed alongside each other.

**Preview window**
Allows the programmer to see a quick representation of how the HTML code will render without having to start a web server to execute the code.

## Types of code editors

There are two types of code editors most commonly used by programmers—lightweight editors and IDEs. The choice of editor to be used depends on the programming language and the type of program to be edited.

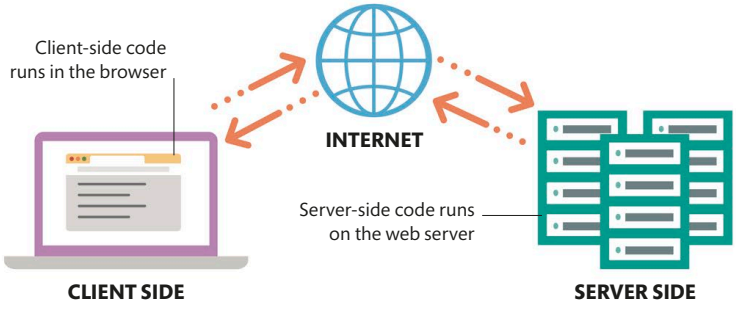
**Lightweight editors**
These editors are used to open and edit a file instantly. They have basic features and are fast and simple to use. Lightweight editors can only be used when working on a single file. This table lists some of the most commonly used lightweight editors.

| LIGHTWEIGHT EDITORS | |
|---|---|
| **Code editors** | **Features** |
| Brackets | An open source code editor that focuses on web development languages, such as HTML, CSS, and JavaScript. It has lots of useful extensions and plugins. (*http://brackets.io/*) |
| Atom | A hackable open source code editor that supports many languages and is designed primarily for web development. Atom is well integrated with Git (a free system for tracking changes in source code) and has lots of custom plugins. (*https://atom.io/*) |
| Sublime Text | A small but powerful code editor that works with several languages and has many tools and shortcuts to aid coding. (*https://www.sublimetext.com/*) |
| Visual Studio Code | Smaller and simpler than the Community edition (see right), Code is a very popular editor that can work with many languages and has advanced features. (*https://visualstudio.microsoft.com/*) |

## CLIENT-SIDE AND SERVER-SIDE SCRIPTING

In client-side scripting, processing takes place in a web browser. The code is transferred from a web server to the user's browser over the internet.

In server-side scripting, processing takes place on a web server. The user sends a request to the web server over the internet, which is fulfilled when the server generates dynamic HTML pages in response and sends them to the user's browser through the same channel.




**Tabs**
Tabs provide an easy way to arrange and manage multiple open files in a code editor. Each tab displays the name of a file, and clicking the name displays the file in a code window.

**Plugins**
Many code editors allow programmers to write plugins to extend the features of a code editor—for example, adding a spell checker or a plugin to format HTML.

**Zoom**
Zooming in makes a part of the text larger and easier to read, while zooming out offers a quick way to view the entire document on the screen in one go.

**Error and warning signs**
These indicate the presence of a spelling mistake or a syntax error that could cause the program to stop executing or behave unexpectedly.

| INTEGRATED DEVELOPMENT ENVIRONMENT | |
|---|---|
| **Code editors** | **Features** |
| WebStorm | A fully featured IDE for web development that uses client-side JavaScript frameworks (see pp.284–285), such as Angular, TypeScript, Vue, and React, and server-side development applications, such as Node.js. (*https://www.jetbrains.com/webstorm/*) |
| NetBeans | Can be used for developing web and desktop applications using open source languages, such as Java and PHP, and web development languages, such as HTML, CSS, and JavaScript. (*https://netbeans.org/*) |
| CodePen | An online code editor that can be used for testing and sharing HTML, CSS, and JavaScript code snippets. It is very useful for finding important components to use on websites. (*https://codepen.io/*) |
| Visual Studio Community | Used to create web and desktop applications for Microsoft, Apple, and Linux environments. It helps programmers build large-scale systems using multiple languages and frameworks. (*https://visualstudio.microsoft.com/*) |

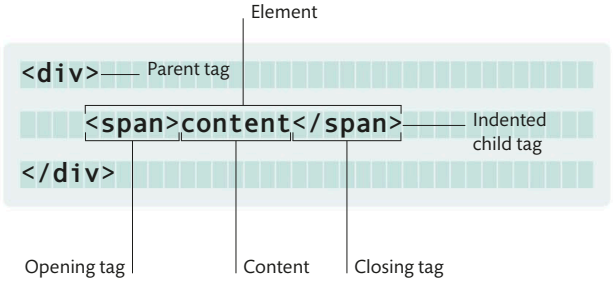
**IDEs**
IDEs are powerful editors that work with many languages and have advanced features that enable a programmer to integrate several languages into a single solution. IDEs are used when working on the entire project. This table lists a few commonly used IDEs.

# Exploring basic HTML

**HTML is the most basic building block of the Web. An HTML file contains all the text, images, and data that is displayed on the browser. It also includes a list of any other files, such as fonts, styles, and scripts that are required to render the HTML elements correctly.**

## HTML tags

An HTML tag is a keyword or a set of characters that defines the format of a web page and determines how its content is displayed onscreen. The combination and order of the HTML tags determine the structure and design of the HTML document. A client browser uses the information in each tag to understand the nature of the tags' content and how to display them correctly. The combination of a tag and its content is known as an element. Some tags, called parent tags, can contain other tags, called children tags. Most tags must have an opening and closing tag, like a set of brackets, but some tags do not require a closing tag and include a closing slash to indicate that they are single tags.




**<body></body>**
The <body> tag contains all the text, data, and images that are displayed when the HTML document is opened in a browser.

** **
This tag contains the text that should appear as a paragraph onscreen. The browser starts a new line and adds margins for spacing around the paragraph.

** **
The  tag is a container for all the HTML elements that can be styled and positioned as a group. This tag displays elements on a new line.

**<img/>**
The <img/> tag is used to describe an image on the page. Its "src" attribute contains the URL that points to the location of the image file.

**<a></a>**
The <a> (anchor) tag describes a hyperlink, which is used to link one page to another. This tag contains the "href" attribute (see p.211), which holds the link's destination.

**<html></html>**
These are outer tags that apply to the entire HTML document. The first <html> tag indicates the markup language used for the document and the </html> tag marks the end of the web page.

**<h1></h1>**
The <h1/h2/h3/h4/h5/h6> tags indicate that the text is a header. <h1> is usually used for the title of the page, while the others are used to style smaller headings on the document.

**<br/>**
The <br/> tag tells the browser to start a new line. It is a single tag, with the closing slash included before the closing greater-than sign.

## INDENTING TAGS

Good programming includes using visual aids to make code more readable. One of the easiest ways to improve the readability of code is to indent child tags inside their parent tags. To help with the indentation, a "Tidy HTML" or "Format HTML" tool can be used to format the code and indent the children tags.

**<head></head>**
The <head> tag contains the metadata that is required to describe the styles, fonts, linked files, page title, and scripts used by the HTML document.

**<title></title>**
This tag contains the text that appears as the title of a document in the browser. There cannot be more than one <title> element in an HTML document.

****
The  tag contains the text and other HTML elements that should appear on the same line.

```
<!doctype HTML>          Document type declaration

<html>                   The header tag
    <head>               Opening <title> tag
        <title> </title>
    </head>              Closing </title> tag
    <body>               The <body> tag
        <p></p>
    </body>
</html>                  The outer </html> tag
```

**HTML document structure**
Every HTML document requires a minimum number of tags. Recognizing the importance of tags in code, most code editors today automatically add these to a blank HTML document.

## Attributes

Most HTML tags have attributes that provide additional information about the HTML element. An attribute describes a property or characteristic of the element. It always appears inside the element's opening tag in a key="value" format. Some attributes may be required by the tag type to render correctly, while other attributes may be optional.

**<img/> tag attributes**
Apart from "src", the "width" and "height" attributes define the dimensions of an image, and the "alt" attribute provides an alternative text description for images that cannot be displayed.

**<a> tag attributes**
The "href" attribute contains a URL that points to the hyperlink's destination, and the "target" attribute instructs the browser to open the hyperlink in a new browser tab or the same tab.

**"id" attribute**
The "id" attribute describes the identity of an element. It can be added to any kind of tag and is specific to it. This attribute can also be used to select the element in CSS and JavaScript.

**"name" attribute**
This attribute is used by input elements to define the name of the property, or characteristic of the element, that is sent to the server. This attribute must be unique to each element in a form.

**"class" attribute**
The "class" attribute describes the name of a group that the element is a part of. Many elements on the same page can be members of the same class.

**"style" attribute**
The "style" attribute describes the visual characteristics of an element. It defines a list of key-value pairs. Each key-value style definition is separated by a semicolon (see p.234).

# HTML forms and hyperlinks

**Web pages are connected by hyperlinks and forms. While hyperlinks send requests for a specific URL, forms send a request that includes data from the current web page. This data is then used by the server to process the request.**

## HTML forms

An HTML <form> tag contains input elements that allow the user to enter data to be sent to the server. When a user clicks the submit button, the browser will send the values of all the input fields in the form to the server. Every input field must have a "name" attribute. This identifier is used as the key for the data value. A form can include various elements for inputting data, including text fields, text areas, labels, checkboxes, radio buttons, select drop-down lists, and hidden fields.

**Labels**
The <label> tag adds a text label to an input control. When the label is clicked, the cursor jumps to the input control. The "for" attribute in the <label> tag must point to the "id" attribute (see p.211) of the input control.

```
<label for="Name">Name:</label>
<input type="text" id="Name" name="Name"
placeholder="Enter name" />
```

**Checkboxes**
This is used for indicating a true or false value. If the checkbox is checked, the browser submits the value in the "value" attribute.

```
<input type="checkbox" name="hasDriversLicense"
value="true"> Do you have a driver's license?
```

**Select drop-down lists**
Select elements allow the user to choose an input from a list of possible values. This selected value is included in the form data sent to the server.

```
<select name="city">
      <option value="delhi">Delhi</option>
      <option value="cairo">Cairo</option>
</select>
```

**Radio buttons**
Radio buttons are used to select one of a group of possible values. Each radio button's "name" attribute will contain the same value. This indicates that they are possible answers for the same field.

```
<input type="radio" name"gender" value="male"
checked/> Male<br/>
<input type="radio" name"gender" value="female"
/> Female
```

## Hyperlinks and URLs

Hyperlinks are text hotspots that, when clicked, navigate the browser to a new HTML document. They can also refer to another element on the same web page, in which case the browser will simply scroll to the required area. In HTML, hyperlinks are indicated by an anchor <a> tag. This tag contains an "href" attribute (see p.211) that stores a URL. This URL is the address to the new HTML document.

**External hyperlink**
These are hyperlinks to an HTML document on another website. It requires a complete URL to navigate.

```
<a href="http://www.dkp.com/otherPage.html">link</a>
```

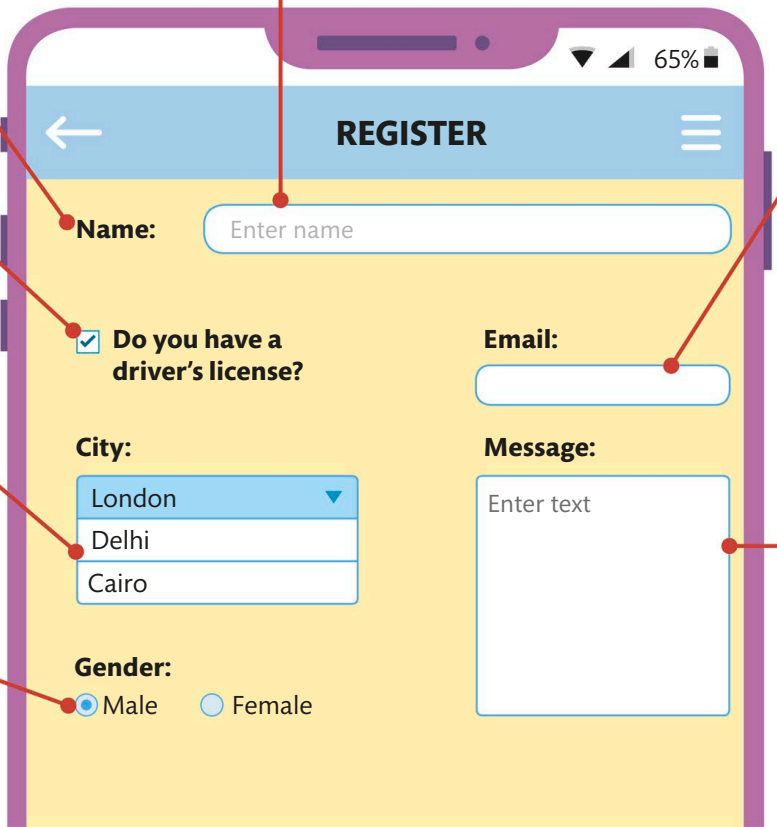
External hyperlinks begin with the "http://" prefix

**Text field**
Text fields are used to enter an alphanumeric value. It is placed on a web page using the <input> tag. The "placeholder" attribute adds a hint to the input text field.

```
<input type="text" name="name"
placeholder="Enter name"/>
```

**Input validation**
Modern browsers use the "type" attribute to help ensure that the correct data is entered in a text input field. Because users can easily enter an invalid value in the browser, input validation must be applied at the server level. Here, the browser will not accept an input unless it is in fact an email address.

```
<label for="email">Email</label>
<input name="emailaddress"
type="email" />
```




**Hidden fields**
These fields do not appear onscreen but will be included in the data sent to the server when the form is submitted. Here, the hidden field could be a unique identity number assigned to the user.

**Text area**
These are text boxes that can accept more than one line of input. Text areas are used to input data that naturally spans multiple lines, such as a paragraph of text or a home address.

```
<textarea rows="5"
cols="40">Enter
text</textarea>
```

# Build a better website

**A well-built website should be easy to read and navigate through. It should be programmed to allow the largest-possible number of clients to view it and should be thoroughly indexed by search engines to draw traffic to the site.**

## Accessibility

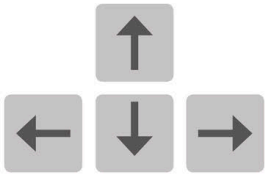
Not all clients are web browsers. An HTML document might also be read on a device that converts the text into braille for the blind or reads the text out loud for people with a hearing disability. An HTML document can be programmed to ensure that it is correctly rendered by these alternative clients. This requires including additional attributes in the HTML tags (see pp.210–211) and adding alternative methods of navigating the site to ensure that it can be accessed by users with special needs. Programmers should think about the topics mentioned below to improve the accessibility of their website.

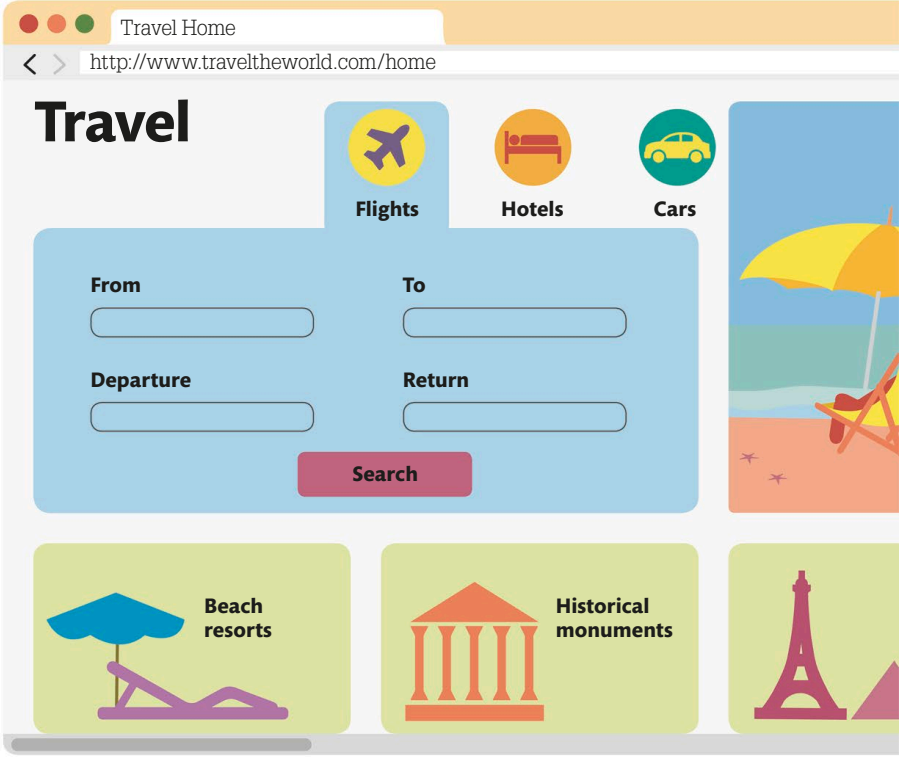
**Readable content**
Ensure there is enough contrast between the background and the text color to make the content easy to read. A dark-colored font will be easier to read on a light background and vice versa.

HELLO WORLD!

**Content organization**
The content of a website should be arranged in a logical and intuitive way. There should be buttons and hyperlinks to suggest the next page the user should visit on their journey through the site. Breadcrumb links show the user where they are in the context of the site and allow them to go back to a previous page if necessary.

**Keyboard alternatives**
Some users may prefer using a keyboard rather than a mouse, so websites should provide for keyboard alternatives for actions, such as scrolling, that usually rely on a mouse.





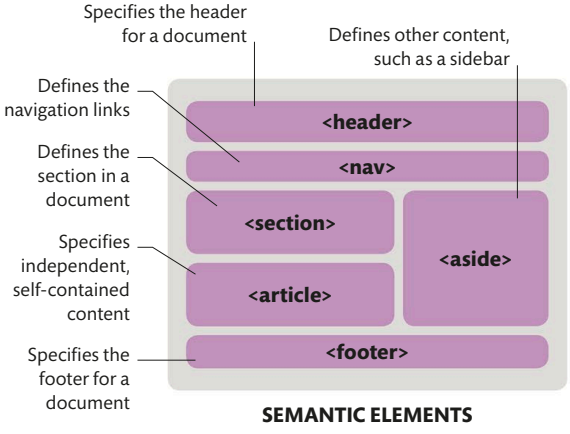

**Text alternatives**
Nonbrowser clients require text alternatives for nontext items. Include an "alt" attribute in an <img/> tag to ensure that such clients can display a text value if they cannot display images.

Describes the image in text

```
<img src="image.jpg" alt="Eiffel Tower"/>
```

## Semantics

One of the key concepts in HTML is that the tags, or semantic elements, should express the meaning of the text, data, and images contained within them. For example, it is expected that an <h1> tag contains the main page header, a <p> tag contains text that should appear in a paragraph style, and a <ul> tag contains items that are all part of a list. Using the correct tag and tag attributes allows browsers and other types of web clients to understand the programmer's intention and correctly render the content in the output format for that client, be it as a web page on a screen or a ticker tape on a braille terminal.




SEMANTIC ELEMENTS


**Responsive layout**
In the past, when the Web was primarily viewed in a browser running on a desktop, the width of HTML documents was commonly defined by a fixed number of pixels. Because many users today view websites on a range of devices, such as smartphones and tablets, it is necessary to code the HTML so that the website can fit on any size screen. The ability to stretch and shape the HTML to fit different screens is known as being "responsive."


TABLET PHONE

**Compliance with guidelines**
All the code should comply with the Web Content Accessibility Guidelines to ensure that users with disabilities are able to enjoy the website. More information can be found at *https://www.w3.org/WAI/standards-guidelines/wcag/*

## Hosting considerations

Web hosting is a service that makes websites accessible over the World Wide Web. Although it is possible to host a website from a personal computer, it is better to do so from a server that is designed to be online 24/7 and can provide backup and security to protect the site.

**Shared hosting**
In shared hosting, the web server hosts many different websites and databases. Each user can rent enough disk space, bandwidth, and database access to provide hosting for a single website.

**VPS (Virtual Private Server) hosting**
This involves a single server being divided into multiple virtual machines. Each website being hosted rents a machine, which is managed as a standalone server, but actually shares resources with all other virtual machines on that server.

**Dedicated server**
A single server is used to host the website, and there is no sharing of resources. The user is responsible for installing and configuring all software and security on the server.

**Elastic cloud computing**
This system can adapt so that the needs of the system match the resources available to it. It provides the most functionality and flexibility but comes at a higher cost than other hosting options.