8.

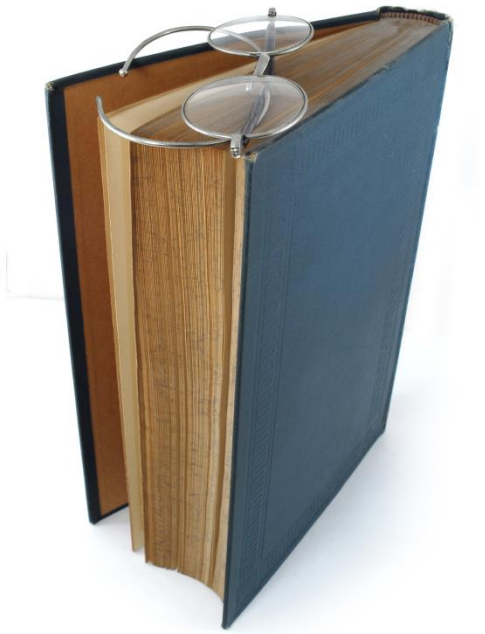
Agenda

- Exploratory Testing
- Networks, IP address and ports
- Client – Server Architecture
  - Two and three-tier applications
- HTTP Protocol
  - Methods, Status Codes
  - Http and Https
- Developer Tool introduction

Common objectives for testing

- **to find bugs**: in the system (dynamic techniques) and documentation (static techniques)
- **to gain confidence:** about level of quality
- **to assess quality:** provide information to stakeholders
- **to prevent future defects:** reviews, early test design, regression testing
- **to address risk:** assess risk, mitigate (reduce) risk
- **to save money:** by finding defects early

What is exploratory testing (ET)?



Exploratory testing is an approach to software testing that is concisely described as simultaneous learning, test design and test execution.

While the software is being tested, the tester learns things that together with experience and creativity generates new good tests to run.

Exploratory testing usually is performed by skilled testers.

Exploratory vs Ad-hoc

Exploratory testing

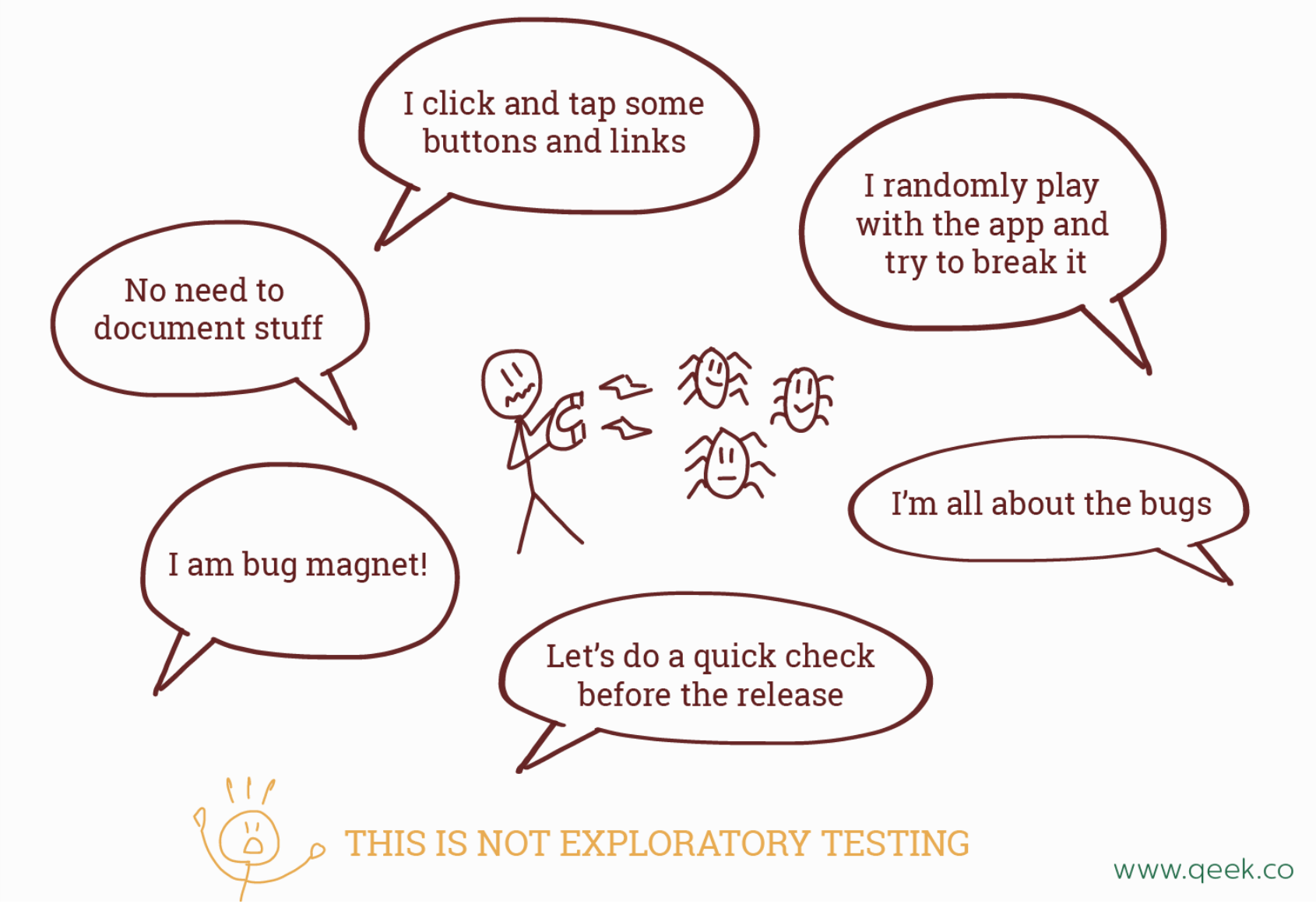
An informal test design technique where the tester actively controls the design of the tests as those tests are performed and uses information gained while testing to design new and better tests

Ad-hoc testing

Testing carried out informally; no formal test preparation takes place, no recognized test design technique is used, there are no expectations for results and arbitrariness guides the test execution activity

Exploratory Testing	Ad-hoc Testing
Aim: to get the information to design new and better tests	**Aim:** to find defects
Result: defects are found and registered; new tests are designed and documented for further usage	**Result:** defects are found and registered

This is not about ET!



Concepts of Exploratory testing

- let's explore, design the tests and test the system concurrently

(James Bach)

- let's learn about the system, test it and report bugs as we go

(Cem Kaner)

- let's apply everything we have learned about testing as we learn about the system, lets do "thinking-while testing"

(James Bach)

- let's design and run tests based upon our experience and knowledge of the system. As our system experience and knowledge increases, more tests are generated

(Lloyd Roden)

Exploratory testing is useful when…

- it is not obvious what the next test should be
- we want to go beyond the obvious tests
- we want to assess the product quickly
- we want to explore areas of the system that are unclear
- creative skills in testing are being encouraged
- we don’t know all the detail of the requirements
- we want to explore areas of the system in more detail
- when we want to avoid “plagiarism”

Doing Exploratory testing

- keep your mission clearly in mind

examples:
- find important bugs first
- provide a general assessment of the product
- find as many bugs as possible in the time given
- gain confidence in the usage of the product
- test for usability and accessibility regulations

- distinguish between testing and observation.

- be aware of the limits of your ability to detect problems.
- keep notes that help you report what you did, why you did it, and support your assessment of product quality.
- keep track of questions and issues raised in your exploration.

ET disadvantages

- ET has no ability to prevent defects
- ET can “waste time” and encourage “bluffing”
- ET is limited to the knowledge and experience of the tester
- ET may only touch a small part of the system and provide false confidence
- ET could promote duplicated testing

ET advantages

- ET is valuable when choosing the next test case: cannot be determined in advance, but can be based on the tester's previous tests
- ET is useful when you need to provide quick feedback on the product's quality at short notice
- ET is useful when we don't have good requirements
- ET is useful when bugs are found in a particular area and we want to delve deeper and provide better information to developers
- ET is useful when you want to "enhance" your existing Test coverage
- ET is useful when we want to train others

Exercise: Testing a multi-function pen

Explore the pen **without unscrewing and breaking it.**

What functions does it offer?

Agenda

- Exploratory Testing
- **Networks, IP address and ports**
- Client – Server Architecture
  - Two and three-tier applications
- HTTP Protocol
  - Methods, Status Codes
  - Http and Https
- Developer Tool introduction

Computer Networks, IP address and ports

A **computer network** is a group of computers that use a set of common communication protocols over digital interconnections for the purpose of sharing resources provided by the network nodes.

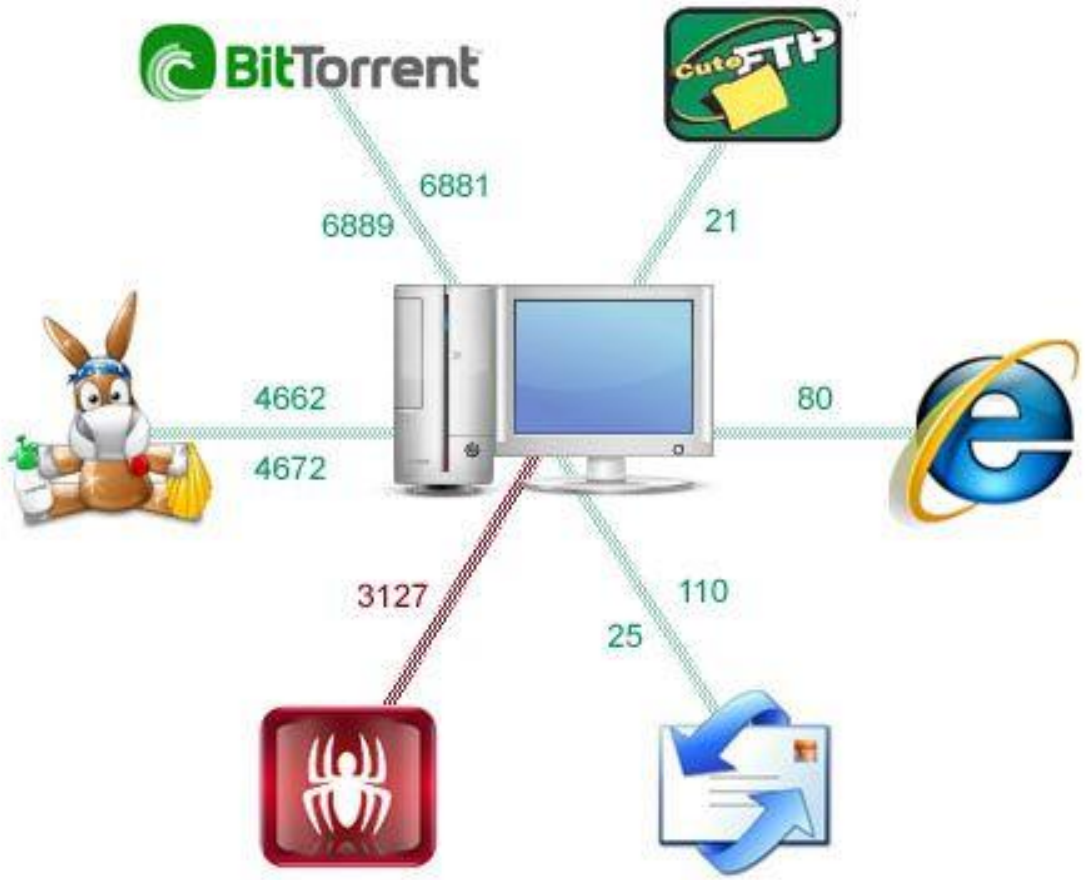
The most popular protocol stack is TCP/IP - a set of network data transfer protocols used in various networks, including the Internet.

The name TCP/IP comes from the two most important protocols. Families - **Transmission Control Protocol (TCP)** and **Internet Protocol (IP)** that were developed and described first in this standard.

Computer Networks, IP address and ports

- **IP (Internet protocol)** - the unique network address of the host in computer network built over IP.

- **Network port** - a conditional number from 1 to 65535, indicating to which application the package is intended.



Computer Networks, IP address and ports

A **server** - is a physical computer dedicated to run services to serve the needs of other computers. Depending on the service that is running, it could be a file server, database server, home media server, print server, or web server.

- listens on a specific port, known in advance to the client
- occupies this port for the whole time until it completes work
- learns about the IP address and port number of the client from invitation sent by client

A Client (or host) - is a computer, connected to other computers for which it provides data or services over a network. In theory, every computer connected to a network acts as a host to other peers on the network. In essence, a host reflects the logical relationship of two or more computers on a network.

- knows the server IP address and port in advance
- selects an arbitrary port that releases after the end of the connection – sends invitation to join

Agenda

- Exploratory Testing
- Networks, IP address and ports
- **Client – Server Architecture**
  - Two and three-tier applications
- HTTP Protocol
  - Methods, Status Codes
  - Http and Https
- Developer Tool introduction

Client - Server Architecture

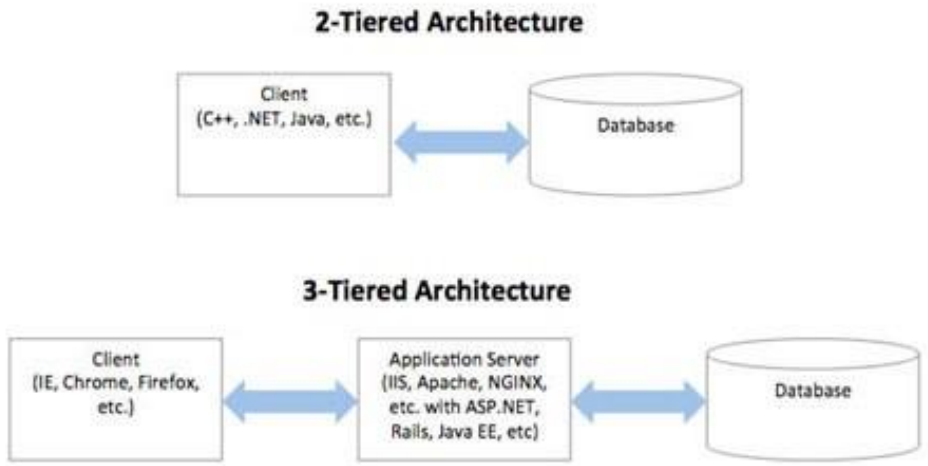
In Computer science, **client-server** is a software architecture model consisting of two parts, client systems and server systems, both communicating over a computer network. A client-server application is a distributed system made up of both client and server software. Client server application provide a better way to share the workload. The client process always initiates a connection to the server, while the server process always waits for requests from any client.

Two and Three-tier network architecture

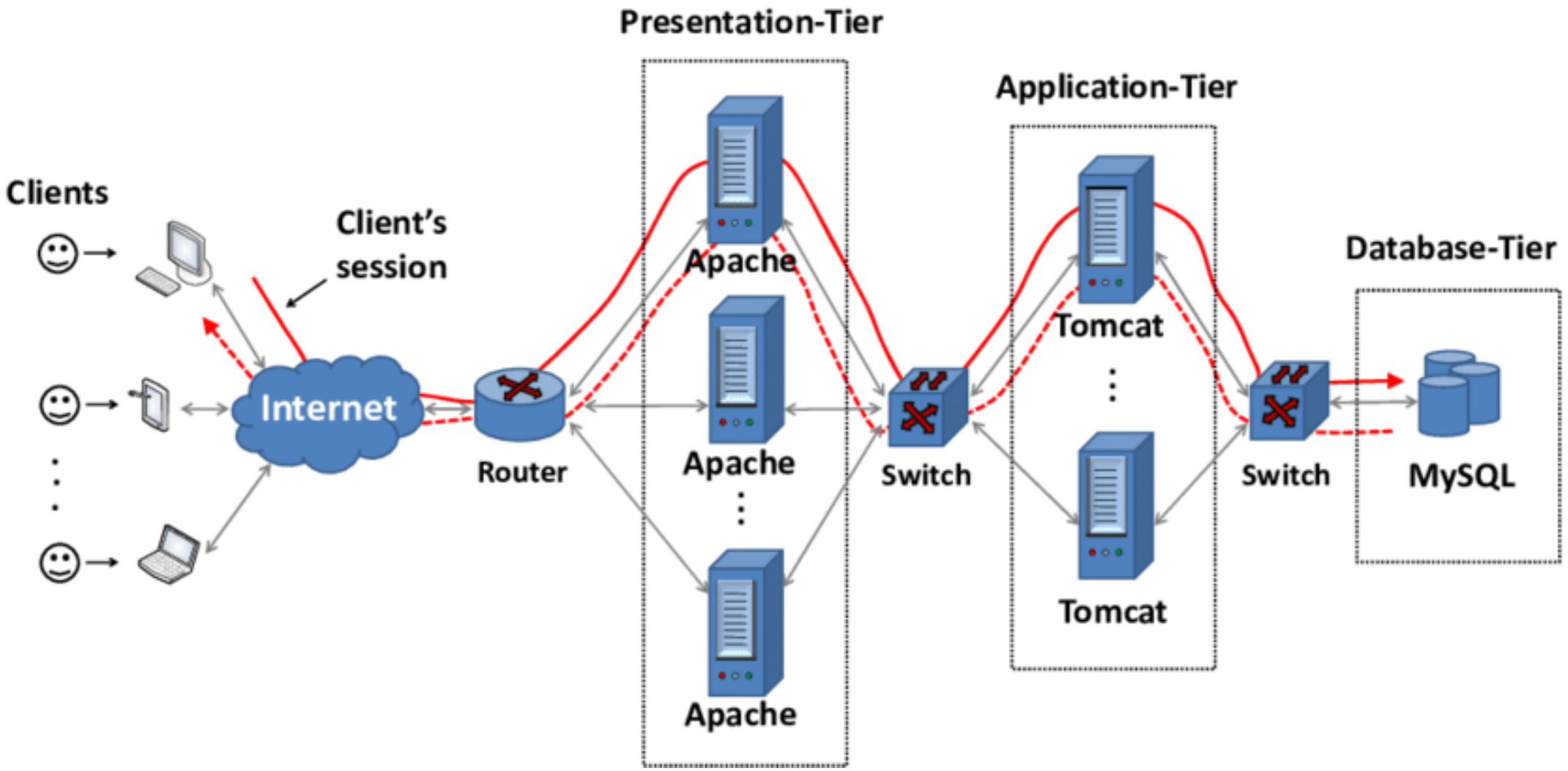
The basic type of **client-server** software architecture employs only two types of hosts: clients and servers. This type of architecture is sometimes referred to as **two-tier**. The **two-tier** architecture means that the client acts as one tier and server process acts as the other tier.

The **client-server** software architecture has become one of the basic models of network computing. Many types of applications have been written using the **client-server** model. Standard networked functions such as E-mail exchange, web access and database access, are based on the client-server model. For example, a web browser is a client program at the user computer that may access information at any web server in the world.

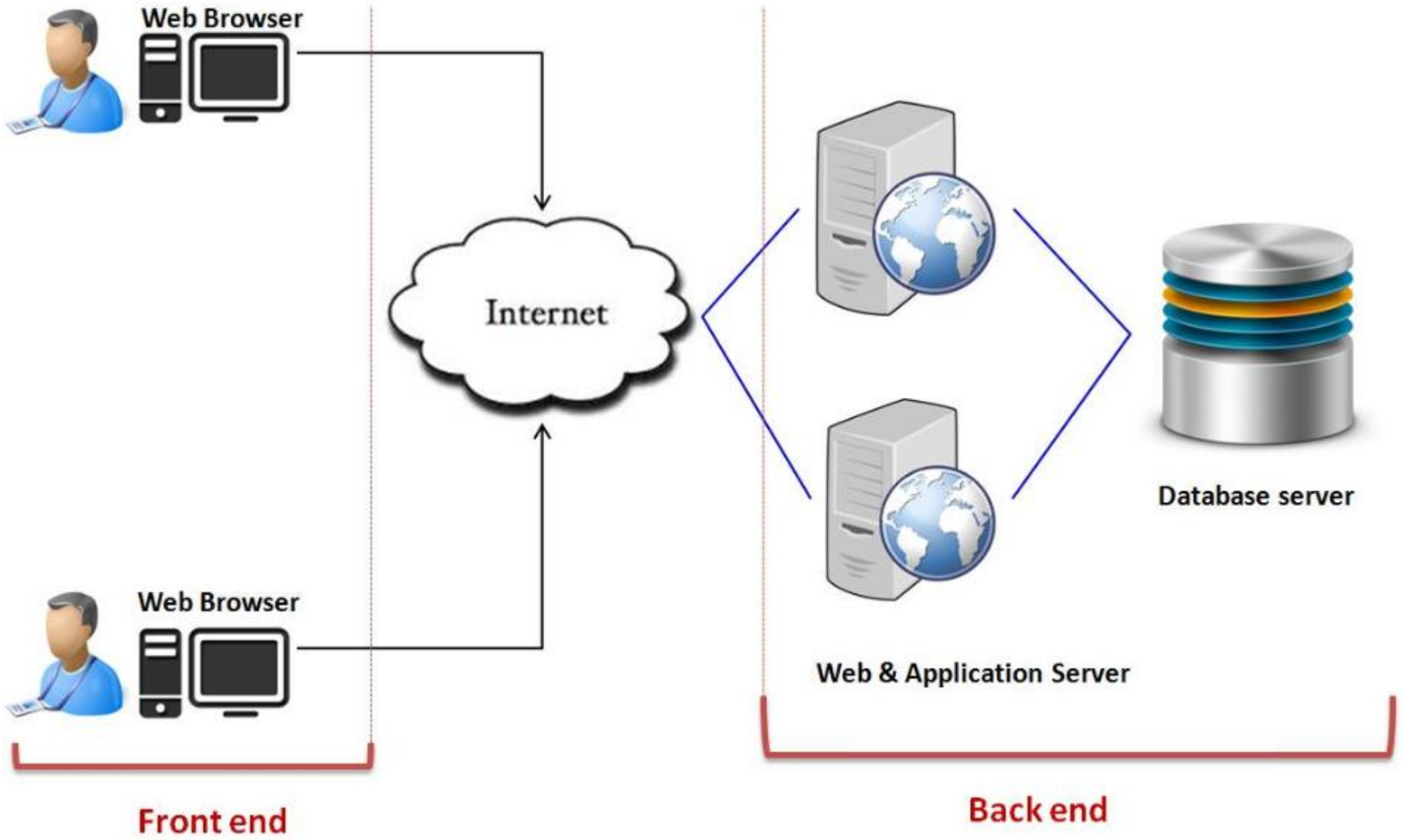
A **three-tier client/server** is a type of multi-tier computing architecture in which an entire application is distributed across three different computing layers or tiers. It divides the presentation, application logic and data processing layers across client and server devices.



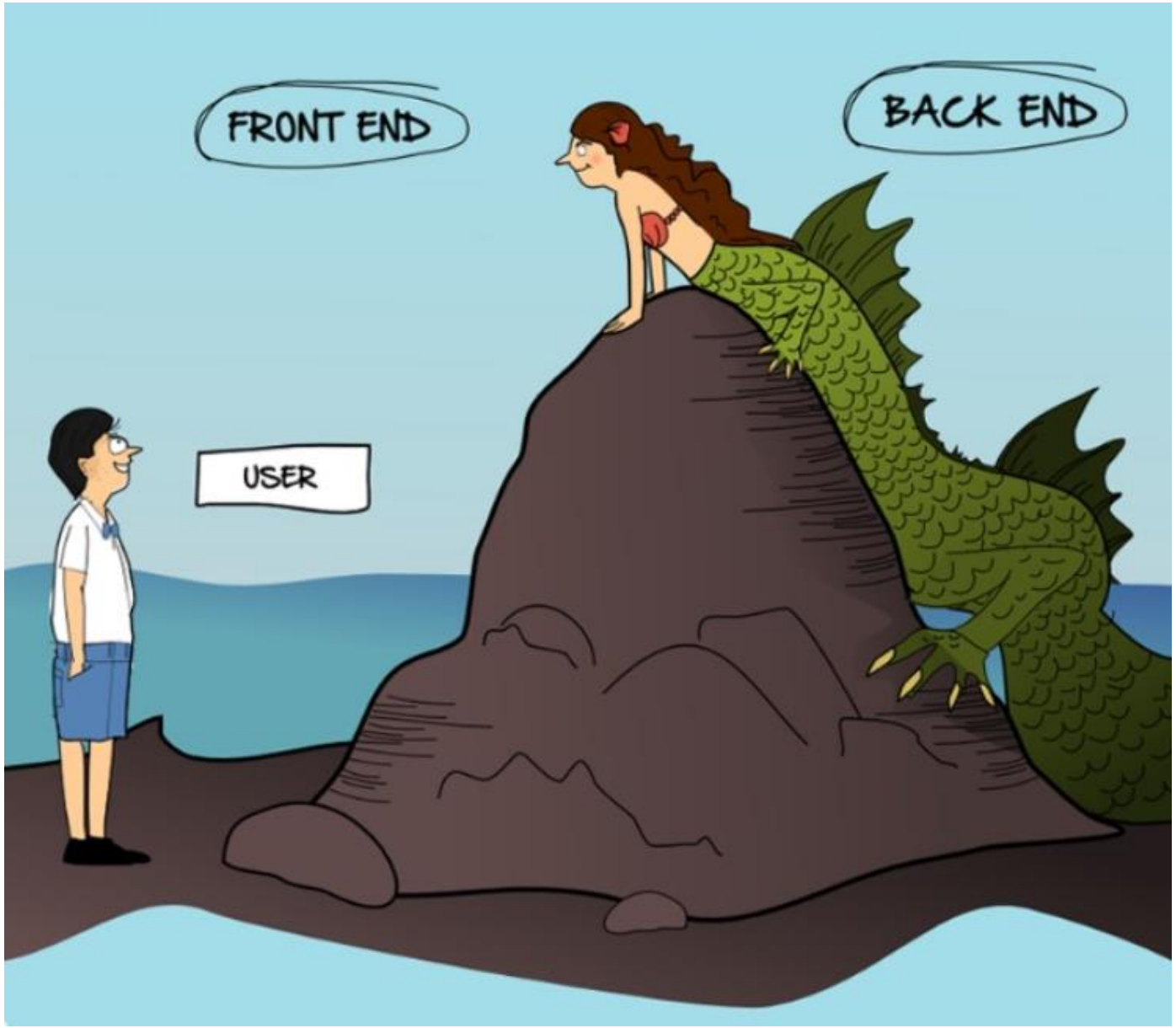
N-tier network architecture



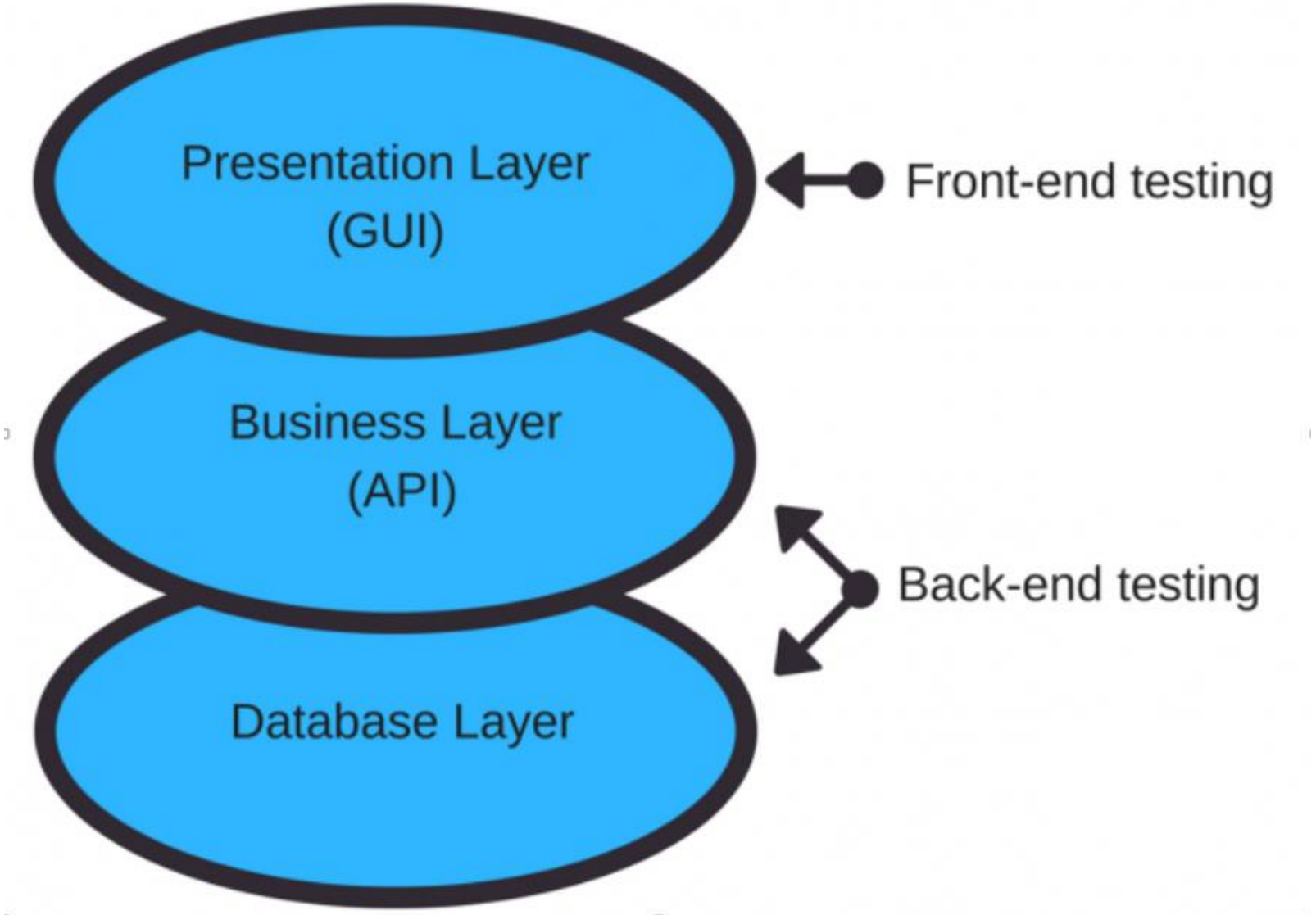
Front-end vs back-end



Front-end vs back-end

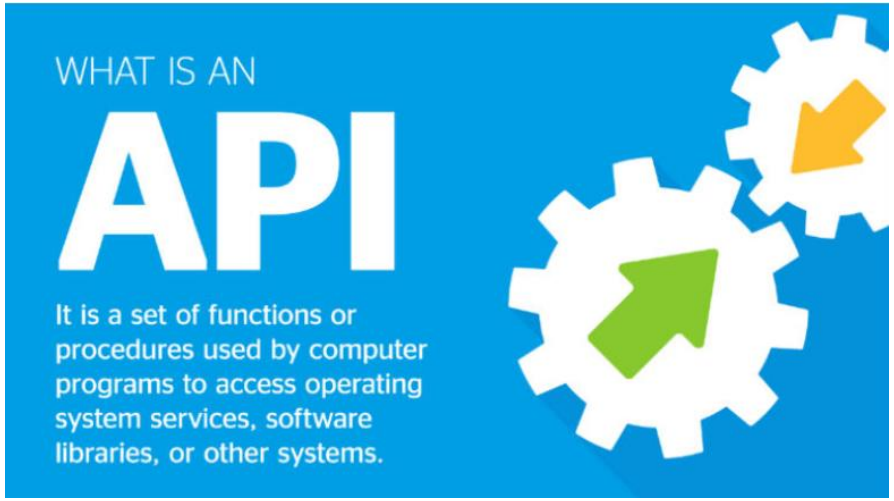


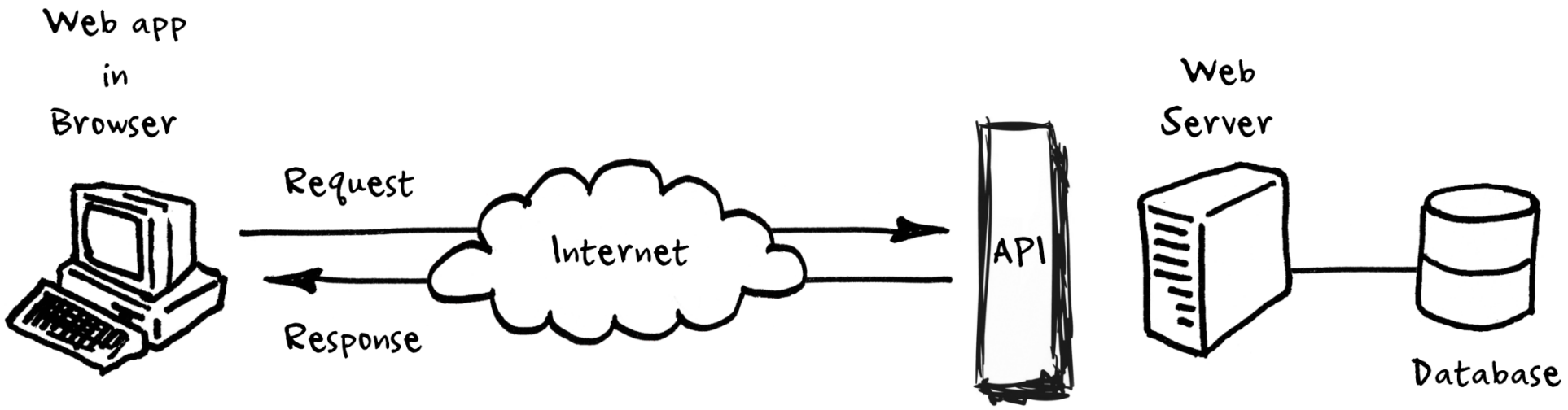
Front-end and back-end testing



Application Programming Interface (API)

Application Programming Interface (API)
In basic terms, APIs just allow applications to communicate with one another. The API is not the database or even the server, it is the code that governs the *access point(s)* for the server applications.





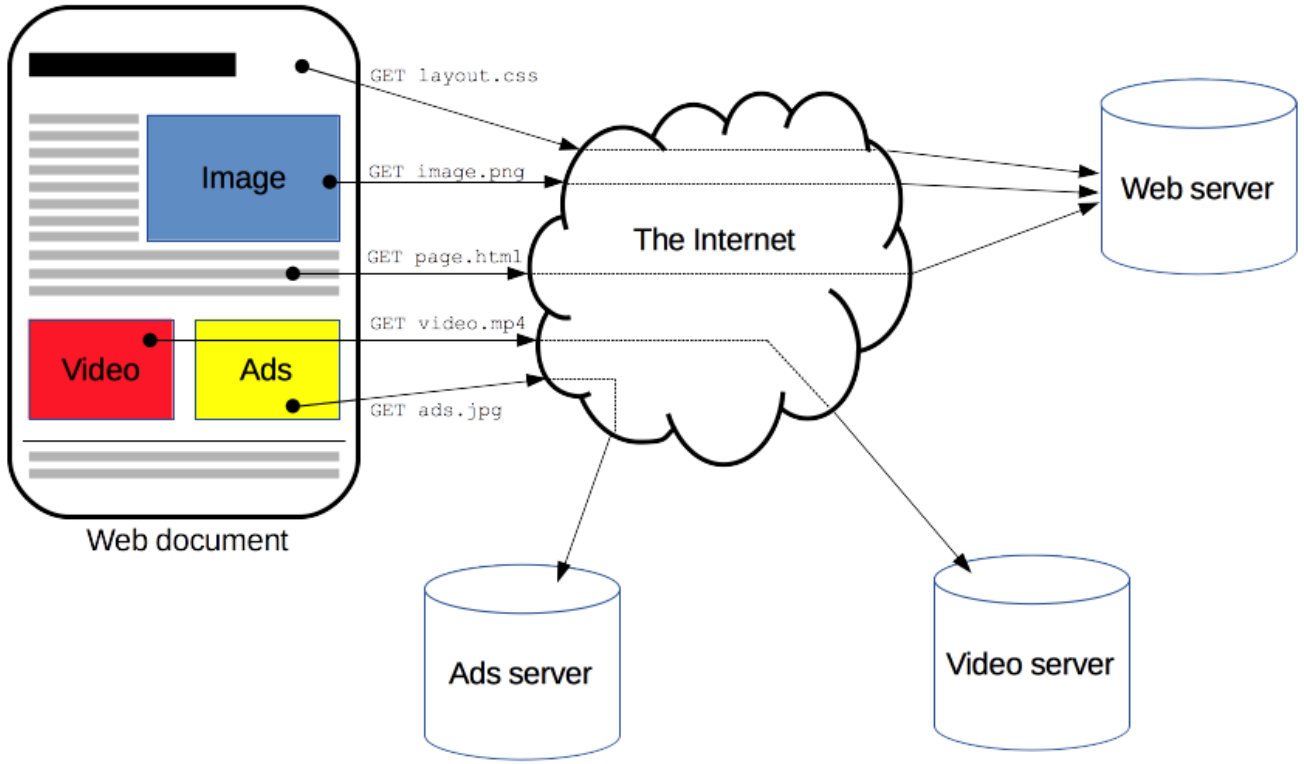
Agenda

- Exploratory Testing
- Networks, IP address and ports
- Client – Server Architecture
  - Two and three-tier applications
- **HTTP Protocol**
  - Methods, Status Codes
  - Http and Https
- Developer Tool introduction

HTTP Protocol

HTTP is an application layer protocol which allows the fetching of resources, such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser.

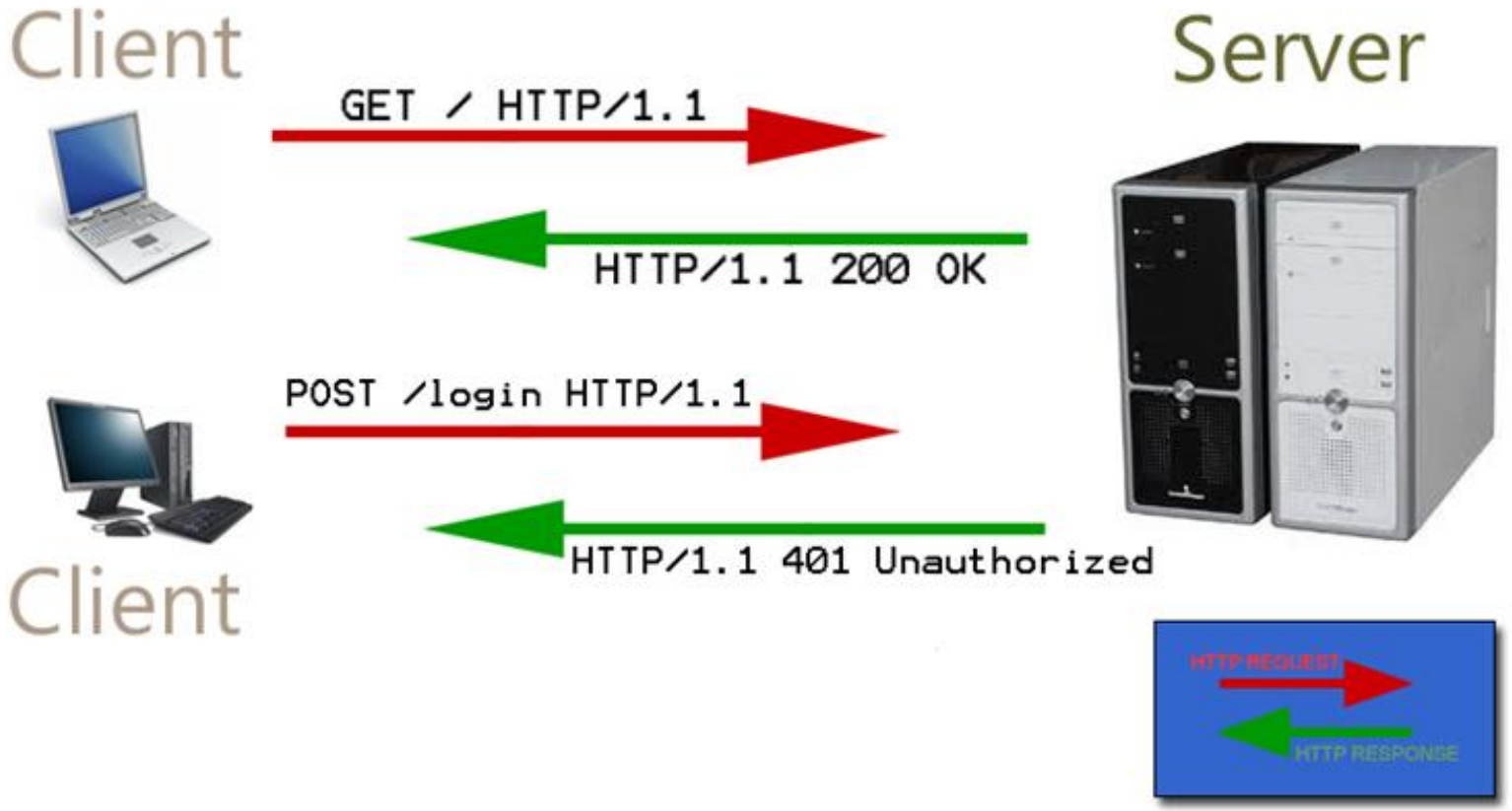
A complete document is reconstructed from the different sub-documents fetched, for instance text, layout description, images, videos, scripts, and more.



HTTP Model

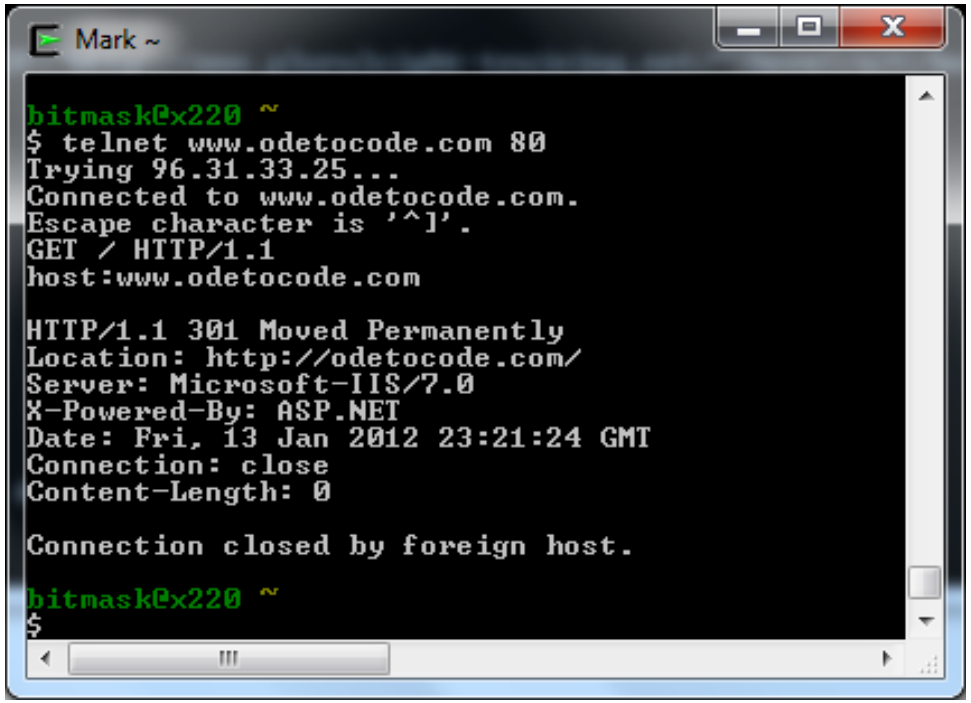
HTTP Request / Response
Communication between clients and servers is done by **requests** and **responses**:

- A client (a browser) sends an **HTTP request** to the web
- A web server receives the request
- The server runs an application to process the request
- The server returns an **HTTP response** (output) to the browser
- The client (the browser) receives the response



HTTP Methods

A web browser knows how to send an HTTP request by opening a network connection and writing out an HTTP message. There is nothing magical about the message – it uses plain ASCII text and a format conforming to the HTTP specification. Any application that can send data over a network can make an HTTP request.

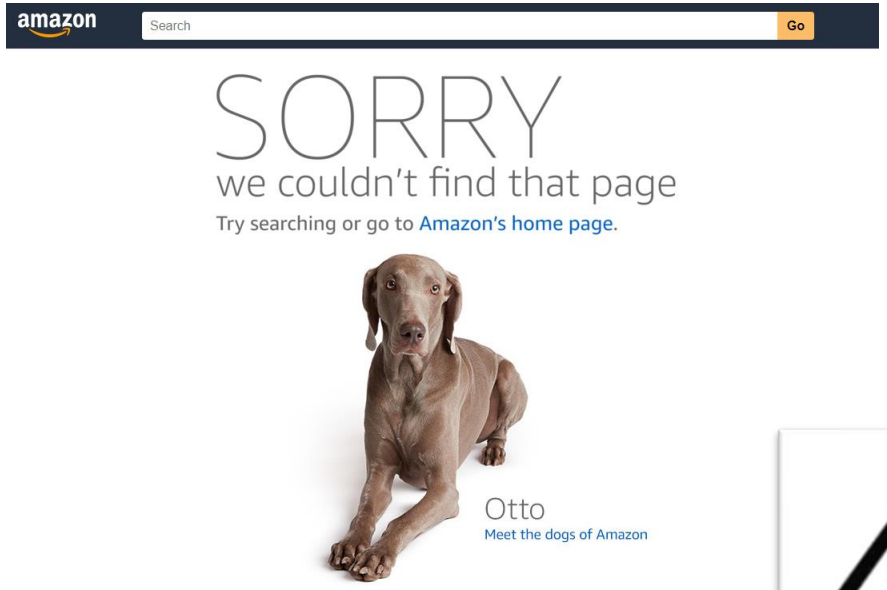


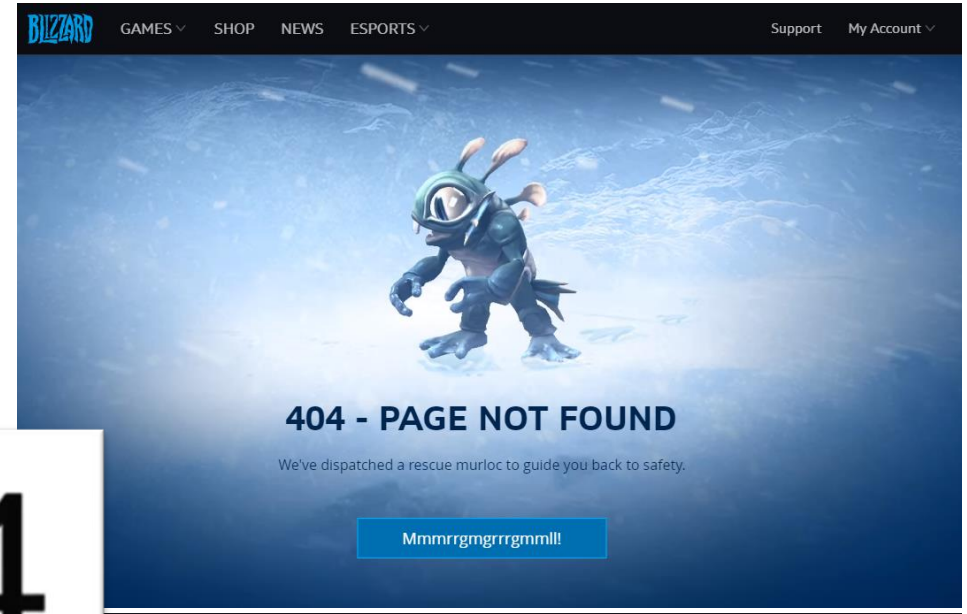
Method	Description
GET	Retrieve a resource
PUT	Store a resource
DELETE	Remove a resource
POST	Update a resource
HEAD	Retrieve just the headers for a resource.

Of these 5 methods, just 2 are the primary workhorses of the web: GET and POST.
A web browser issues a GET request when it wants to retrieve a resource like a page, an image, a video, or a document.
GET requests are the most common type of request.

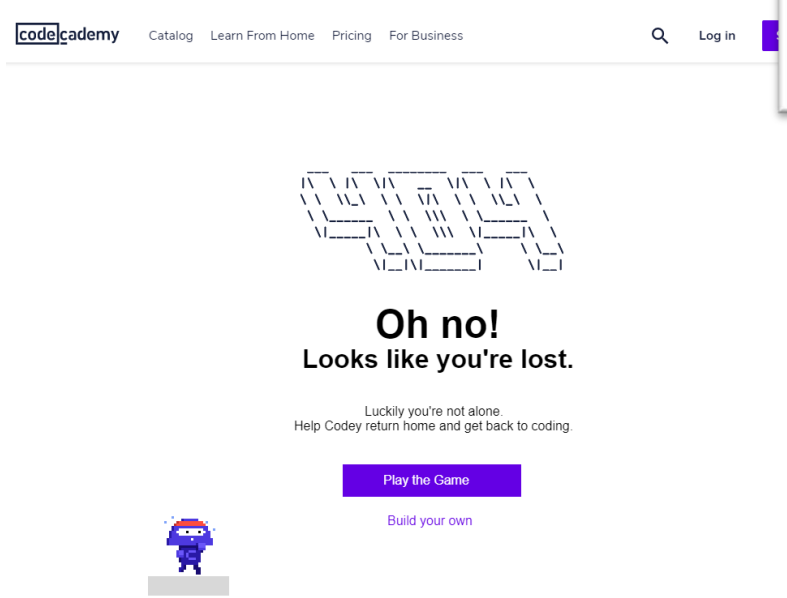
HTTP response status codes

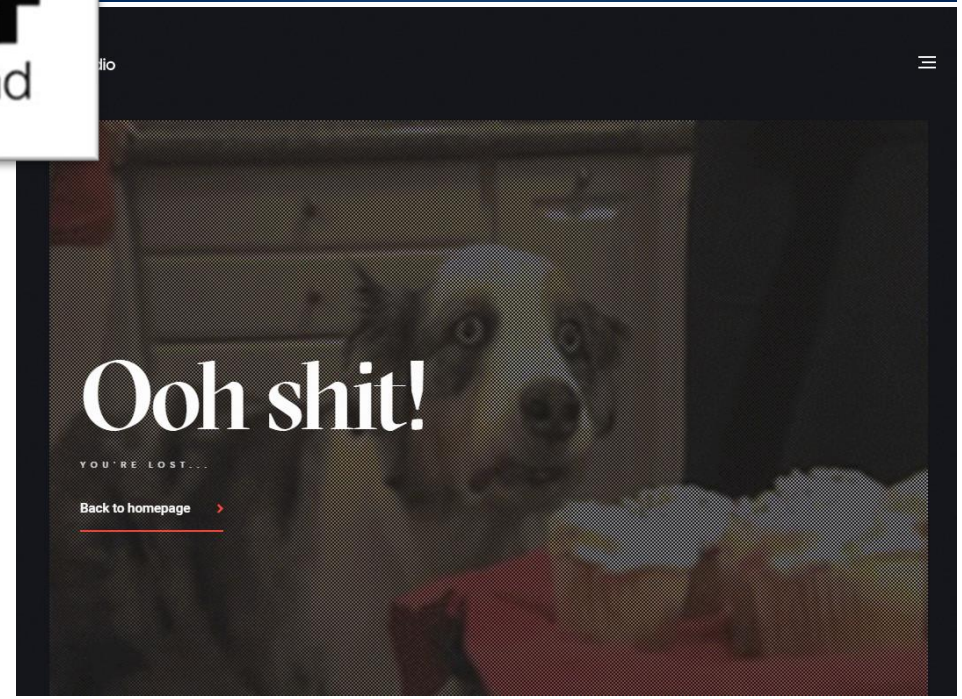
Type	Status Codes	Examples
Informational	1xx	100 Continue, 101 Switching Protocols
Success	2xx	200 - OK , 201 - Created, 202 Accepted
Redirection	3xx	300 Multiple Choices, 301 Moved Permanently, 302 - Found
Client Error	4xx	400 Bad Request, 403 - Forbidden , 404 - Not Found , 422 - Unprocessable Entity
Server Error	5xx	500 - Internal Server Error , 503 - Service Unavailable


amazon
Search
Go
SORRY
we couldn't find that page
Try searching or go to Amazon's home page.
Otto
Meet the dogs of Amazon

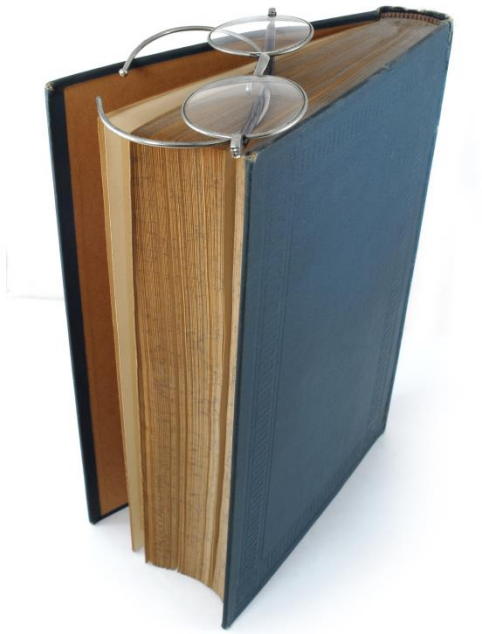

BLIZZARD
GAMES
SHOP
NEWS
ESPORTS
Support
My Account
404 - PAGE NOT FOUND
We've dispatched a rescue murloc to guide you back to safety.
Mmmrrgmgrrrgmmll!


404
Page not found


codecademy
Catalog
Learn From Home
Pricing
For Business
Log in
Oh no!
Looks like you're lost.
Luckily you're not alone.
Help Codey return home and get back to coding.
Play the Game
Build your own

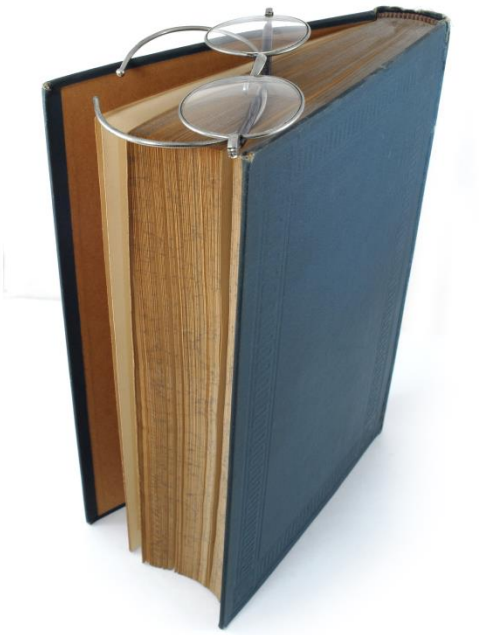

Ooh shit!
YOU'RE LOST...
Back to homepage

What is a website?



A **site** or **website** is a central location of various web pages that are all related and can be accessed by visiting the home page of the website using a browser (e.g., Internet Explorer, Edge, Safari, Firefox, or Chrome).

What is a webpage?



A **web page** or **webpage** is a document commonly written in HyperText Markup Language (HTML) that is accessible through the Internet or other network using an Internet browser. A web page is accessed by entering a URL address and may contain text, graphics, and hyperlinks to other web pages and files.

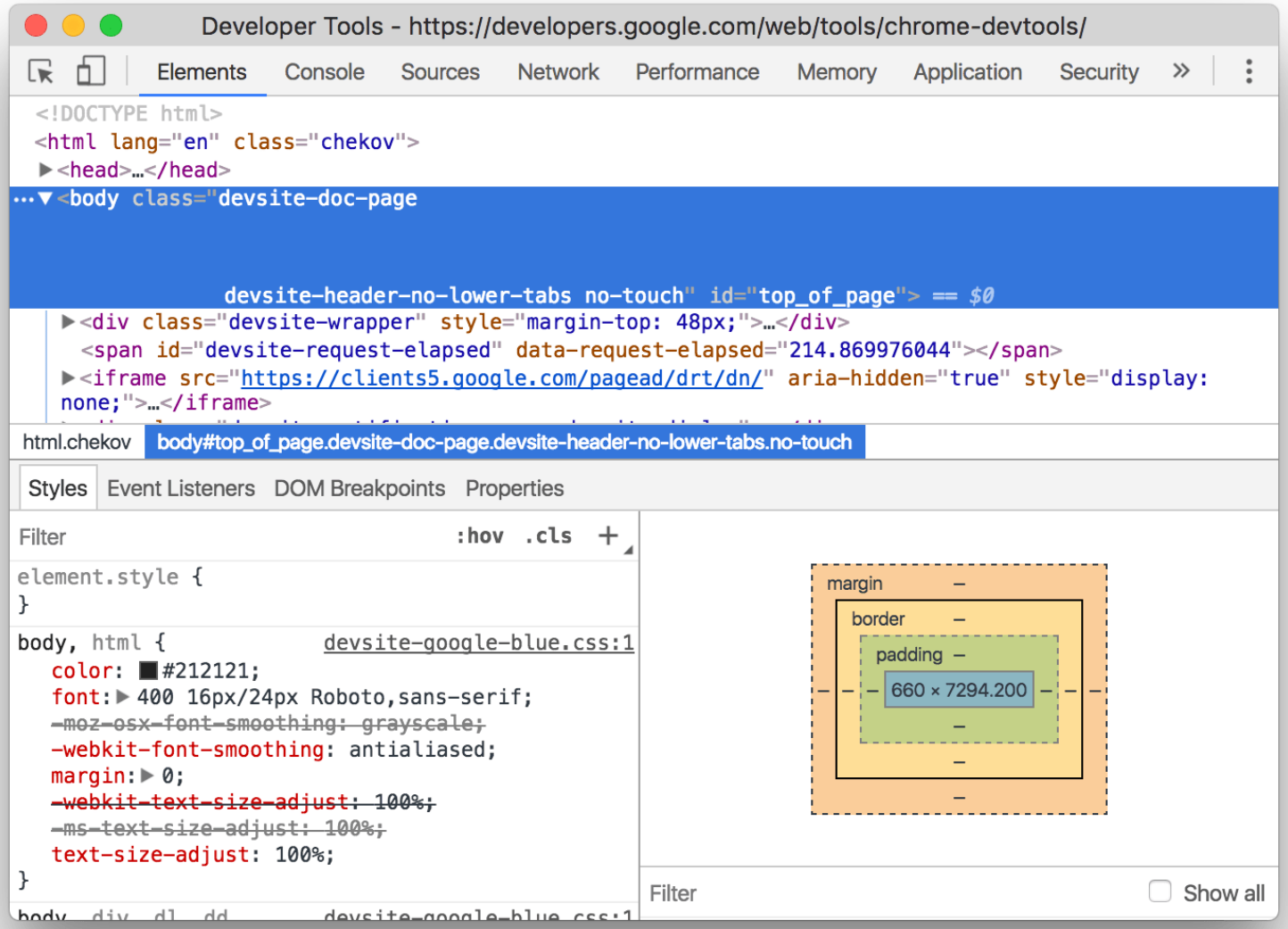
Example of HTML – JavaGuru page

```
<!doctype html>
<!--[if (gte IE 9)|!(IE)]><!-->
<html lang="ru-RU" class="js_active  vc_desktop  vc_transform  vc_transform">
  <!--<![endif]-->
▶ <head>…</head>
▼ <body class="home page-template-default page page-id-137 tribe-js has-topbar-w
transparent-header-w wpb-js-composer js-comp-ver-4.12.1 vc_responsive">
  ▶ <div id="wrap" class=" colorskin-custom  online-t ">…</div>
    <!-- end-wrap -->
    <!-- End Document
    ================================================== -->
  ▶ <script>…</script>
  ▶ <script>…</script>
  ▶ <script type="text/javascript">…</script>
    <script type="text/javascript" src="https://javaguru.lv/wp-content/plugins/contact-form-
    7/includes/js/scripts.js?ver=5.0.4"></script>
    <script type="text/javascript" src="https://javaguru.lv/wp-content/themes/michigan/js/
    jquery.plugins.js"></script>
  ▶ <script type="text/javascript">…</script>
    <script type="text/javascript" src="https://javaguru.lv/wp-includes/js/mediaelement/
    mediaelement-and-player.min.js?ver=4.2.6-78496d1"></script>
    <script type="text/javascript" src="https://javaguru.lv/wp-includes/js/mediaelement/
    mediaelement-migrate.min.js?ver=4.9.14"></script>
  ▶ <script type="text/javascript">…</script>
    <script type="text/javascript" src="https://javaguru.lv/wp-content/themes/michigan/js/
    jquery.masonry.min.js"></script>
    <script type="text/javascript">
    /* <![CDATA[ */
    var scrolls_value = "330";
    /* ]]> */
    </script>
    <script type="text/javascript" src="https://javaguru.lv/wp-content/themes/michigan/js/
```

Agenda

- Exploratory Testing
- Networks, IP address and ports
- Client – Server Architecture
  - Two and three-tier applications
- HTTP Protocol
  - Methods, Status Codes
  - Http and Https
- **Developer Tool introduction**

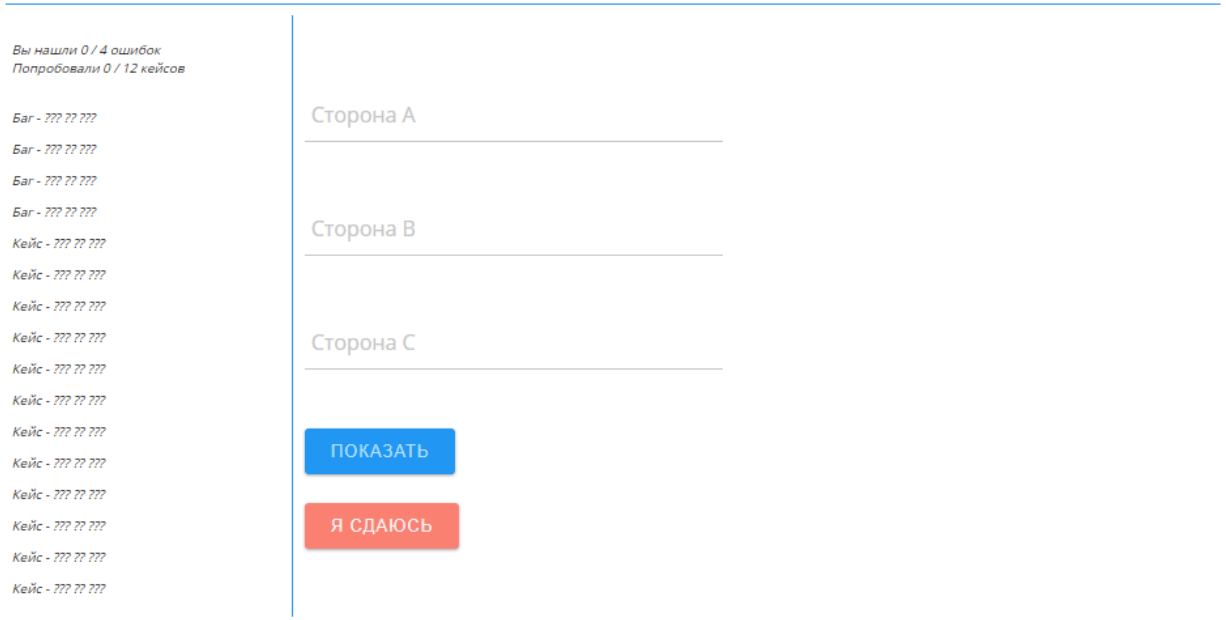
Developer Tools



Homework

Explore the Triangle Puzzle.

Try to find bugs and all possible triangle cases.



Remember the triangle rule: sum of any 2 sides of a triangle must be greater than the third side.