Databases and Cloud Concepts Notes

paraphrased by Tyler Wright

An important note, these notes are absolutely ***NOT*** *guaranteed to be correct, representative of the course, or rigorous. Any result of this is not the author's fault.*

Contents

1 The Internet

The internet is a world-wide computer network, connecting computing devices also known as hosts or end systems. These connections can take many forms, such as cables and radio waves. Intermediate switching devices inbetween hosts are known as routers.

1.1 Clients and Servers

A program or machine that responds to requests from others is called a server. A program or machine that sends requests to a server is a client.

1.2 Internet Layers

There are four internet layers:

Layer	Common Protocol	Description
Application	HTTP	Web browsers making requests and parsing responses
Transport	TCP	Breaks requests down into numbered packets and can reassemble messages
Network	IP	Attaches addresses to packets and groups packets based on their incoming addresses
Physical		Sends bits to the local router and assembles bits into packets

1.3 Protocols

Protocols are an agreement on how to communicate.

1.3.1 HTTP - HyperText Transfer Protocol

There are four main operations that can be carried out on HTTP resources:

Operation	Performed by...
Creation	`HTTP POST`
Reading	`HTTP GET`
Updating	`HTTP PUT`
Deletion	`HTTP DELETE`

Requests are formed by an operation as well as a `host` and `content-type` parameter to describe the format of information.

1.3.2 URL - Uniform Resource Locator

Each URL is formed by a scheme (like `http` or `https`), a host (like `www.bristol.ac`), a path (like `.uk/home/maths`). Paths can have queries attached, preceded by `?` as parameters.

2 HTML - HyperText Markup Language

2.1 Tags, Attributes and, Values

Tags form the structure of HTML, with `html`, `head` and, `body` usually forming the top levels:

```
<html>
    <head>
        <title>Title<\title>
    <\head>
    <body>
        <p>Paragraph<\p>
    <\body>
<\html>
```

Attributes form parts of tags and, as expected, assign attributes to tags. This can describe the width of elements (`width`), the hyperlink attached to text (`href`) and, more:

```
<a href="www.bristol.ac.uk">Bristol<\a>
```

2.1.1 Common Tags

Below is a table containing common HTML tags:

Tag	Description
`h1`, ... `h6`	Headings
`p`	Paragraph
`br`	New line
`ul`	Unordered list
`ol`	Ordered list
`li`	List item
`em`	Emphasis
`strong`	Importance
`q`	Quote
`cite`	Citation
`var`	Variable
`code`	Source code

2.2 Block and Inline Elements

Block level elements take up the full width of the container and start on new lines, so stack vertically.
Inline elements don't start on new lines and only take up as much width as is necessary, so stack horizontally.

2.2.1 Block Tags

Below is a table containing some of the block HTML tags:

Tag	Description
`header`	This is the very top of the page
`main`	This fills the space inbetween the header and footer
`section`	This forms subsections of blocks
`div`	No meaning, for layout purposes
`p`	This forms paragraphs of text
`figure`	This forms images
`nav`	This fills the space left of the main block
`aside`	This fills the space right of the main block
`footer`	This is the very bottom of the page

2.3 Common Attributes

Below is a table containing some of the common HTML attributes:

Attribute	Description
`id`	Uniquely identifies the tag with the value
`class`	Marks tags you want to operate as a group

2.4 Forms

The form tag, shown in the example:

```
<form method="post" action="/comment/comment.php">
    <p>
        <label for="name">Name:</label>
        <input type="text" id="name" />
    </p>
    <p>
        <button type="submit">OK</button>
    </p>
</form>
```

The `method` attribute takes two values `GET` and `POST`. The former places the information in the URL parameters and the latter utilises a HTTP request.
The `action` attribute defines an action to be performed when the form is submitted. In this case, it's sent to a PHP script.
The `label for` attribute should link to a `input id`. Additionally, the `input name` attribute is the key which accompanies the input value in the request.
The `button` tag has three types, a `submit` button that makes the request, a `reset` button that resets all fields and, a `button` type that does nothing by default but can be configured using Javascript.

Types can be used to make field use a specific format or be required. Additionally, they can be given placeholder text and autocompletion properties.

2.5 Escape Characters

We list these below, note that they also work in XML:

Character	HTML Representation
<	`<`
>	`>`
&	`&`
”	`&quot;`
Non-breaking Space	` `

3 CSS - Cascading Stylesheets

CSS describes how HTML elements should be drawn to the screen. It can be used:

- Inline with the `style` attribute,
- Internally with the `style` tag in the `head` section,
- Externally via linking to a `.css` file.

3.1 Stylesheet Linking

We can link to external stylesheets as follows:

```
<link rel="stylesheet" href="styles.css">
```

3.2 CSS File Structure

The parts of CSS files are formed as follows:

```
selector {
    key: value;
}
```

3.2.1 Selectors

Selectors can be a:

- tag, written simply as `div`,
- class, written as `.class`,
- id, written as `#id`.

4 Encoding

Encoding is about mapping symbols to bytes. There are many standards, of which we will see a few.

4.1 ASCII - American Standard Code for Information Interchange

ASCII contains the digits 0 to 9, the lowercase and uppercase English alphabet, some punctuation and, special characters. Each of these is represented by a seven bits.

4.2 UTF - Unicode Transformation Format

The first 128 characters of UTF-8 correspond to the characters of ASCII making UTF-8 backwards compatible with ASCII. The unicode character set contains around 136,000 characters. The individual formats (UTF-8, UTF-16, etc.) encode these differently and have different memory requirements. We can choose to use UTF-8 in HTML as follows:

```
<meta charset="UTF8" />
```

4.3 CSV - Comma Separated Values

CSV use commas to separate field and `CR LF` to separate records. The record at the top is reserved (usually) for the titles of the columns.

4.3.1 Streams

We can read CSV files in as streams. Thinking about stream operations is important when considering web programming as data is usually a stream. We cannot perform operations on streams that require more than one pass (like standard deviation).

A few operations we can do are:

- filter - omitting as we go,
- map - mapping as we go,
- sum - summing as we go.

5 Representing Data

5.1 Trees

Representing data as trees requires three separators for the start and end of an item, and for fields. Additional separators are needed for quoting and escaping.

5.2 XML - Extensible Markup Language

The goal of XML is to create a straight-forward way of representing data that is machine and human readable that also can give context to data.

It allows portable, non-proprietary, hierarchical data storage. Common parsers are XPath and XQuery.

5.2.1 The Structure

XML documents are formed of five components:

- the XML declaration,
- the root element,
- attributes,
- child elements,
- text data,

illustrated below:

```
<?xml version="1.0" encoding="UTF-8"?>
<labReport patientId="1234567890" specimenID="750853">
    <testResult date="2005-01-25-T12:15:37-09:00">
        <test>
            <testCode scheme="myCodes">42Hxx</testCode>
            <testName>Potassium</testName>
        </test>
    </testResult>
</labReport>
```

5.2.2 Validation

There are two validation methods, DTD (Document Type Definition) and schema. We consider the example:

```
<candidate>
    <name>Catherine Slade</name>
    <party>
        <name>Green</name>
    </party>
    <ward>
        <name>Bedminster</name>
        <electorate>9951</electorate>
    </ward>
</candidate>
```

we have the DTD validation format:

```
<?xml version="1.0"?>
<!DOCTYPE candidate [
    <!ELEMENT candidate (name, party, ward)>
    <!ELEMENT name (#PCDATA)>
    <!ELEMENT party (name)>
    <!ELEMENT ward (name, electorate)>
    <!ELEMENT electorate (#PCDATA)>
]>
<candidate> ... </candidate>
```

where PCDATA is parsed character data. Also, we have the XML Schema Definition:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="candidate">
        <xs:complexType><xs:sequence>
    <xs:element name="name" type="xs:string" />
    <xs:element name="party"><xs:complexType><xs:sequence>
        <xs:element name="name" type="xs:string" />
    </xs:sequence></xs:complexType></xs:element>
    <xs:element name="ward"><xs:complexType><xs:sequence>
        <xs:element name="name" type="xs:string" />
        <xs:element name="electorate"
                    type="xs:nonNegativeInteger" />
        </xs:sequence></xs:complexType></xs:element>
    </xs:sequence></xs:complexType></xs:element>
</xs:schema>
```

5.3 JSON - Javascript Object Notation

JSON is a machine and human friendly data format. As it is formed by text, it can be parsed and generated by most programming languages and can be transmitted easily.

5.3.1 Comparisions to XML

Here are some key differences:

- JSON allows arrays,
- JSON tends to be shorter,
- JSON is quicker to read and write,
- JSON can be parsed by standard functions.

6 Databases

6.1 Web Architecture

A multitier architecture or n-tier architecture is a client-server architecture which physically separates presentation, application processing and data management functions.

A common example is the 3-tier architecture which is formed by presentation, application and, database layers.