



มหาวิทยาลัยขอนแก่น

จ.ขอนแก่น



KHON KAEN UNIVERSITY

Advanced HTML

Assoc. Prof. Dr. Kanda Runapongsa Saikaew

(krunapon@kku.ac.th)

Department of Computer Engineering
Khon Kaen University



คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น
FACULTY OF ENGINEERING KHON KAEN UNIVERSITY

Agenda

- Hyperlinks
- Documents and website structure
- Debugging HTML
- Multimedia and embedding
- HTML tables
- HTML forms



Block level links

- You can turn just about any content into a link, even block level elements
- If you have an image you wanted to turn into a link, you could just put the image between `<a>` tags
- Example

```
<a href="http://kku.ac.th">  
      
</a>
```



Document fragments

- It is possible to link to a specific part of an HTML document
 - To do this you first have to assign an id attribute to the element you want to link to a specific heading
 - Example

```
<h2 id="Mailing_address">Mailing address</h2>  
<p>Want to write us a letter? Use our <a  
 href="contacts.html#Mailing_address">mailing  
 address</a>.</p>  
<p>The <a href="#Mailing_address">company mailing  
 address</a> can be found at the bottom of this page.</p>
```

Link best practices

- Use clear link wording
 - Visual readers skim over the page rather than reading every word, and their eyes will be drawn to page features that stand out, like links
 - They will find descriptive link text useful

Good link text: [Download Firefox](#)

```
1 <p><a href="https://firefox.com/">
2   Download Firefox
3 </a></p>
```

Bad link text: [Click here to download Firefox](#)

```
1 <p><a href="https://firefox.com/">
2   Click here
3 </a>
4 to download Firefox</p>
```

Other tips

- Don't repeat the URL as part of the text
 - URLs look ugly, and sound even uglier when a screen reader reads them out letter by letter
- Don't say “link” or “links to” in the link text
- Keep your link label as short as possible
- Minimize instances where multiple copies of the same text are linked to different places



Use relative links wherever possible

- You should use relative links wherever possible when linking to other locations within the *same website*
 - For a start, it is a lot easier to scan your code — relative URLs are generally a lot shorter than absolute URLs, which makes reading code much easier
 - With a relative URL on the other hand, the browser just looks up the file that is being requested, on the same server

Linking to non-HTML resources

- When linking to a resource that will be downloaded (like a PDF or Word document) or streamed (like video or audio), you should add clear wording to reduce any confusion

```
1 <p><a href="http://www.example.com/large-report.pdf">
2   Download the sales report (PDF, 10MB)
3 </a></p>
4
5 <p><a href="http://www.example.com/video-stream/" target="_blank">
6   Watch the video (stream opens in separate tab, HD quality)
7 </a></p>
8
9 <p><a href="http://www.example.com/car-game">
10  Play the car game (requires Flash)
11 </a></p>
```



Use the download attribute when linking to a download

- When you are linking to a resource that is to be downloaded rather than opened in the browser, you can use the download attribute to provide a default save filename

```
1 <a href="https://download.mozilla.org/?product=firefox-latest-ssl&os=win64&lang=en-US"
2   download="firefox-latest-64bit-installer.exe">
3     Download Latest Firefox for Windows (64-bit) (English, US)
4   </a>
```



E-mail links

- It is possible to create links or buttons that, when clicked, open a new outgoing email message rather than linking to a resource or a page
- This is done using the `<a>` element and the mailto: URL scheme

```
1 | <a href="mailto:nowhere@mozilla.org">Send email to nowhere</a>
```



Basic sections of a document (1/2)

- header
 - This is where the main common information about a website usually stays from one webpage to another
- navigation bar
 - Links to the site's main sections; usually represented by menu buttons, links, or tabs
 - Like the header, this content usually remains consistent from one webpage to another

Basic sections of a document (2/2)

- main content
 - A big area in the center that contains most of the unique content of a given web page
- sidebar
 - Some peripheral info, links, quotes, ads, etc. Usually this is contextual to what is contained in the main content
- footer
 - A strip across the bottom of the page that generally contains fine print, copyright notices, or contact info



Sample sections of a document

Header

[HOME](#) [OUR TEAM](#) [PROJECTS](#) [CONTACT](#) Gol

Article heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a diam lectus. Sed sit amet ipsum mauris. Maecenas congue ligula ac quam viverra nec consectetur ante hendrerit. Donec et mollis dolor. Praesent et diam eget libero egestas mattis sit amet vitae augue. Nam lacinia congue enim, ut porta lorem lacinia consectetur.

subsection

Donec ut libero sed accu vehicula ultricies a non tortor. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ut gravida lorem. Ut turpis felis, pulvinar a semper sed, adipiscing id dolor.

Pellentesque auditor nisi id magna consequat sagittis. Curabitur dapibus, enim sit amet elit pharetra tincidunt feugiat nisl imperdiet. Ut convallis libero in urna ultrices accumsan. Donec sed odio eros.

Another subsection

Donec viverra mi quis quam pulvinar at malesuada arcu rhoncus. Cum sociis natoque penatibus et manibus parturient montes, nascetur ridiculus mus. In rutrum accumsan ultricies. Mauris vitae nisi at sem facilisis semper ac in est.

Vivamus fermentum semper porta. Nunc diam velit, adipiscing ut tristique vitae sagittis vel odio. Maecenas convallis ullamcorper ultricies. Curabitur ornare, ligula semper consectetur sagittis, nisi diam iaculis velit, et fringilla sem nunc vel mi.

Related

- [Oh I do like to be beside the seaside](#)
- [Oh I do like to be beside the sea](#)
- [Although in the North of England](#)
- [It never stops raining](#)
- [Oh well...](#)

©Copyright 2050 by nobody. All rights reversed.



Tags representing sections

- header: <header>
- navigation bar: <nav>
- main content: <main> with various content subsections represented by <article>, <section>, and <div> elements
- sidebar: <aside>; often placed inside <main>
- footer: <footer>



Sample HTML sections

- <https://krunapon.github.io/sections.html>

The screenshot shows a web browser window with the URL <https://krunapon.github.io/sections.html> in the address bar. The page has a header with the word "Header" in bold. Below the header is a navigation bar with links for HOME, OUR TEAM, PROJECTS, and CONTACT, along with a search bar and a "Go!" button. The main content area contains several sections: "Header 2" with two paragraphs, "Header 3" with three paragraphs, and another "Header 3" with two paragraphs. To the right, there is a sidebar titled "Related" containing a bulleted list of five items, each with a blue link.

Header

HOME OUR TEAM PROJECTS CONTACT

Search query Go!

Header 2

Paragraph

Header 3

Paragraph

Paragraph

Header 3

Paragraph

Paragraph

Related

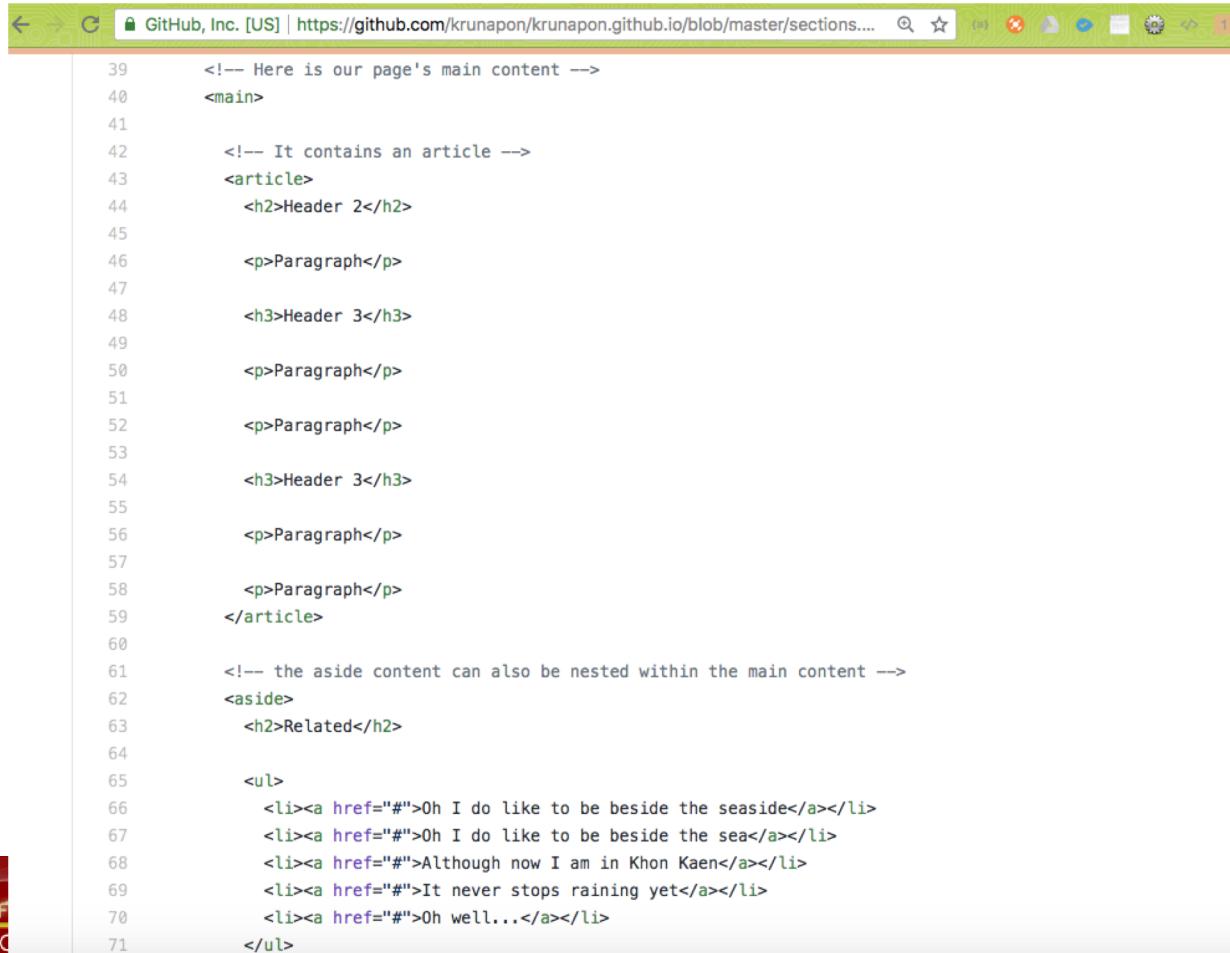
- [Oh I do like to be beside the seaside](#)
- [Oh I do like to be beside the sea](#)
- [Although now I am in Khon Kaen](#)
- [It never stops raining yet](#)
- [Oh well...](#)

©Copyright 2018 by nobody. All rights reversed.



File sections.html

<https://github.com/krunapon/krunapon.github.io/blob/master/sections.html>

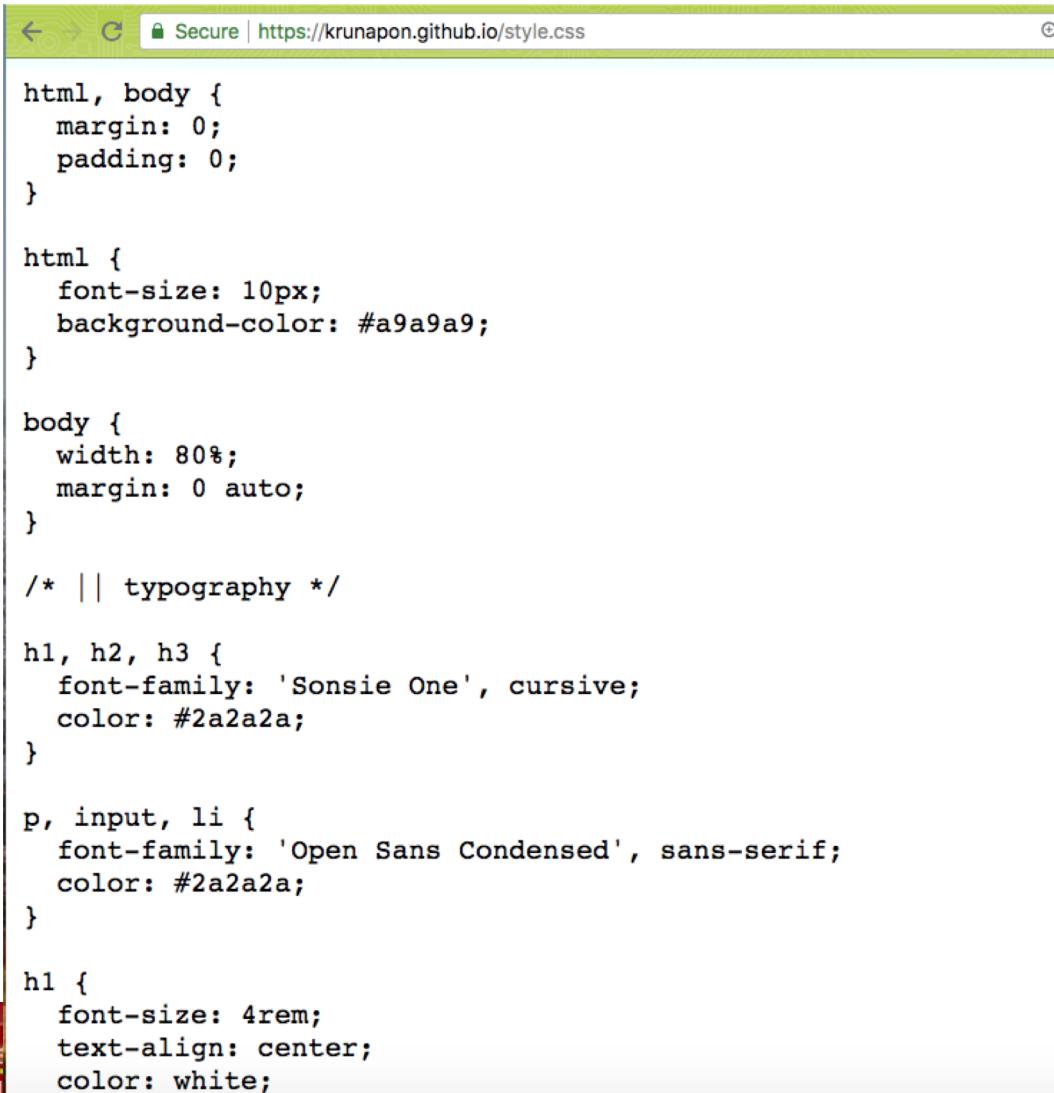


The screenshot shows a GitHub code editor displaying the file `sections.html`. The code is written in HTML and includes several nested sections and an aside element.

```
39     <!-- Here is our page's main content -->
40     <main>
41
42         <!-- It contains an article -->
43         <article>
44             <h2>Header 2</h2>
45
46             <p>Paragraph</p>
47
48             <h3>Header 3</h3>
49
50             <p>Paragraph</p>
51
52             <p>Paragraph</p>
53
54             <h3>Header 3</h3>
55
56             <p>Paragraph</p>
57
58             <p>Paragraph</p>
59         </article>
60
61         <!-- the aside content can also be nested within the main content -->
62         <aside>
63             <h2>Related</h2>
64
65             <ul>
66                 <li><a href="#">Oh I do like to be beside the seaside</a></li>
67                 <li><a href="#">Oh I do like to be beside the sea</a></li>
68                 <li><a href="#">Although now I am in Khon Kaen</a></li>
69                 <li><a href="#">It never stops raining yet</a></li>
70                 <li><a href="#">Oh well...</a></li>
71             </ul>
```

File style.css

- <https://krunapon.github.io/style.css>



```
Secure | https://krunapon.github.io/style.css

html, body {
    margin: 0;
    padding: 0;
}

html {
    font-size: 10px;
    background-color: #a9a9a9;
}

body {
    width: 80%;
    margin: 0 auto;
}

/* || typography */

h1, h2, h3 {
    font-family: 'Sonsie One', cursive;
    color: #2a2a2a;
}

p, input, li {
    font-family: 'Open Sans Condensed', sans-serif;
    color: #2a2a2a;
}

h1 {
    font-size: 4rem;
    text-align: center;
    color: white;
}
```

Line breaks and horizontal rules

- Two elements that you'll use occasionally and will want to know about are `
` and `<hr>`
- `
` creates a line break in a paragraph; it is the only way to force a rigid structure in a situation where you want a series of fixed short lines
- `<hr>` elements create a horizontal rule in the document that denotes a thematic change in the text (such as a change in topic or scene).



Example: Line breaks & horizontal rules

The screenshot shows a web browser window with the URL <https://krunapon.github.io/sections.html>. The page content includes:

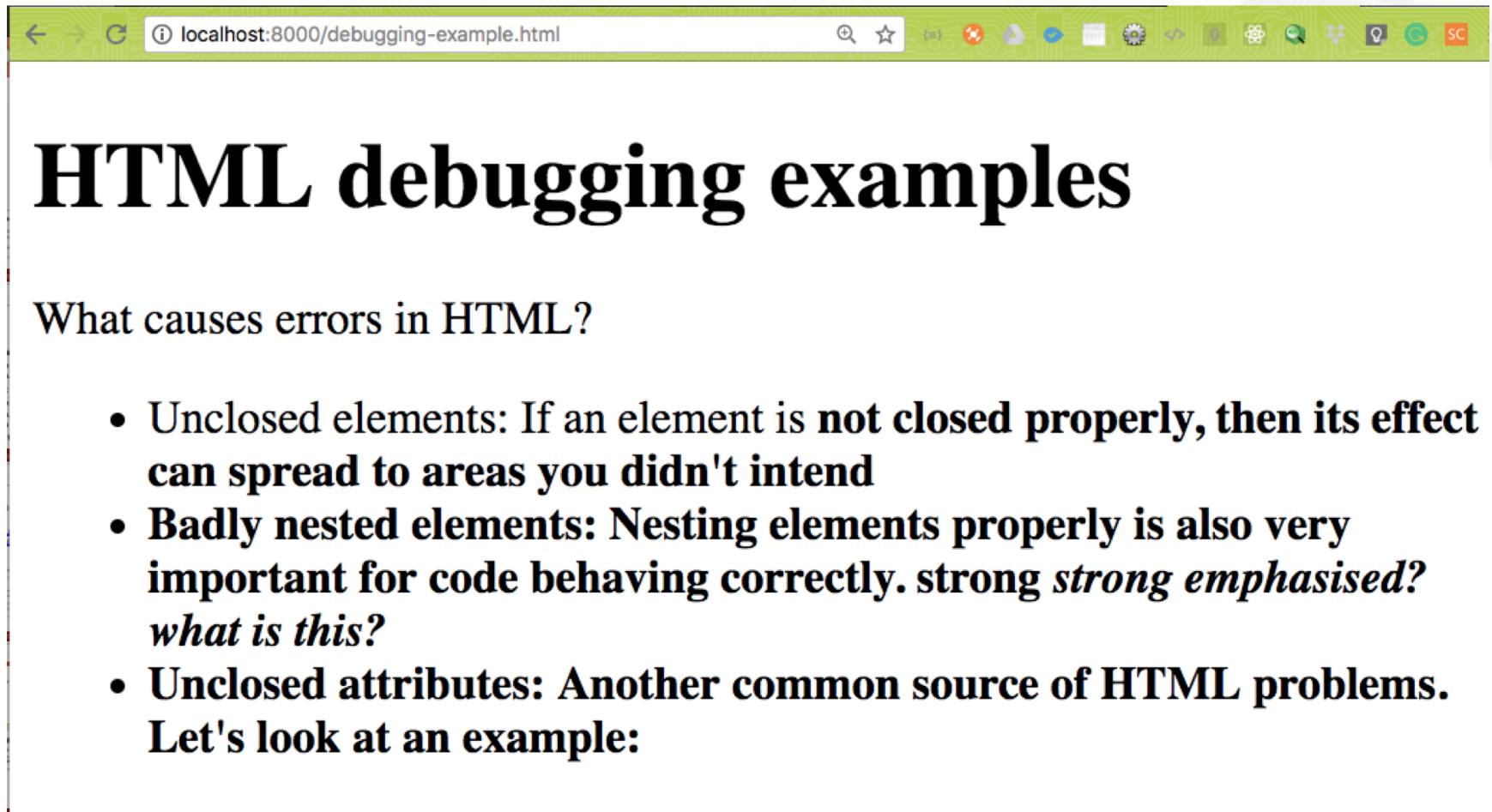
- A large **Header 2** section.
- A **Paragraph** section.
- A **Header 3** section.
- A **Paragraph** section containing text: "Paragraph. This is an example of br
The line after br".
- A **Paragraph** section containing text: "Paragraph. This is an example of br
The line after hr".
- A pink sidebar titled **Related** containing a bulleted list:
 - [Oh I do like to be beside the seaside](#)
 - [Oh I do like to be beside the sea](#)
 - [Although now I am in Khon Kaen](#)
 - [It never stops raining yet](#)



Errors in HTML

- **Syntax errors**
 - These are spelling errors in your code that actually cause the program not to run
- **Logic errors**
 - These are errors where the syntax is actually correct, but the code is not what you intended it to be, meaning that program runs incorrectly

HTML debugging example



The screenshot shows a web browser window with a green header bar. The address bar displays 'localhost:8000/debugging-example.html'. The main content area contains the text 'HTML debugging examples' in a large, bold, black serif font.

HTML debugging examples

What causes errors in HTML?

- Unclosed elements: If an element is **not closed properly, then its effect can spread to areas you didn't intend**
- Badly nested elements: Nesting elements properly is also very important for code behaving correctly. **strong *strong emphasised? what is this?***
- Unclosed attributes: Another common source of HTML problems. **Let's look at an example:**



HTML Debugging example code

<h1>HTML debugging examples</h1>

<p>What causes errors in HTML?

Unclosed elements: If an element is

not closed properly, then its effect can spread to areas you didn't intend

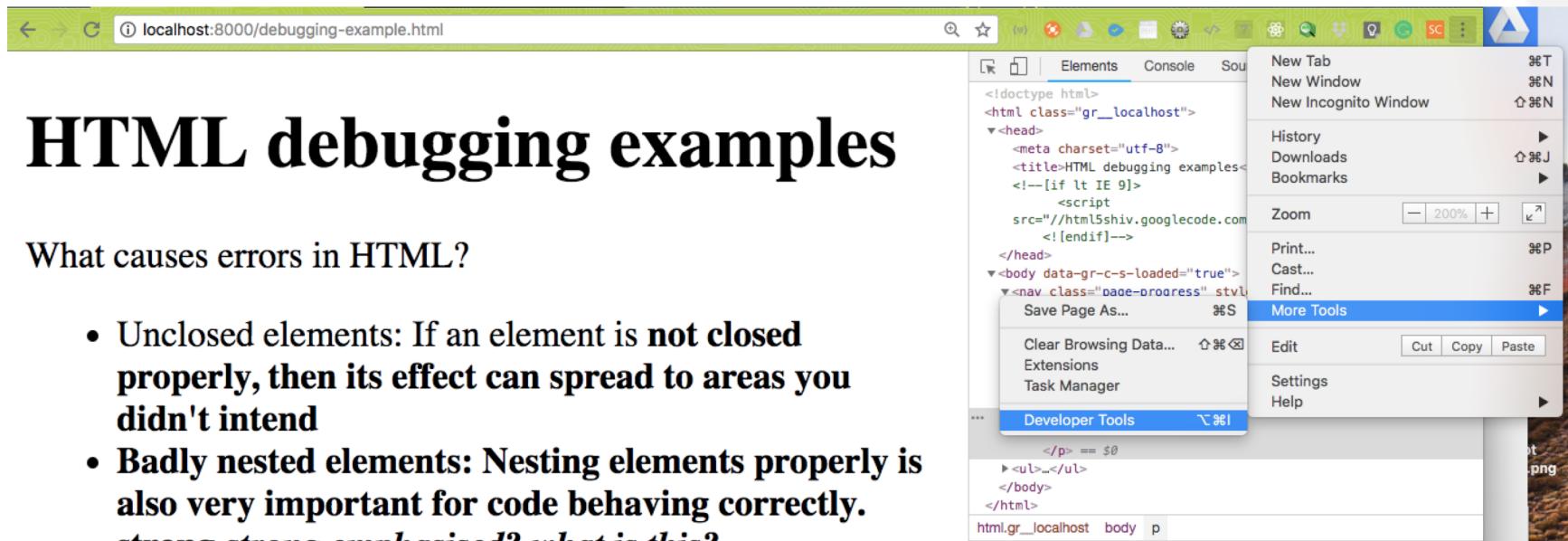
Badly nested elements: Nesting elements properly is also very important for code behaving correctly.

strong strong emphasised? what is this?

Unclosed attributes: Another common source of HTML problems link to Mozilla homepage



Opening the developer tool



A screenshot of a web browser window displaying the URL `localhost:8000/debugging-example.html`. The page content is titled "HTML debugging examples". On the right side of the browser, the developer tools menu is open, showing various options like New Tab, History, and Developer Tools. The Developer Tools option is highlighted with a blue background.

HTML debugging examples

What causes errors in HTML?

- Unclosed elements: If an element is **not closed properly, then its effect can spread to areas you didn't intend**
- Badly nested elements: Nesting elements properly is also very important for code behaving correctly.**
strong strong emphasized? what is this?

```
<!doctype html>
<html class="gr__localhost">
  <head>
    <meta charset="utf-8">
    <title>HTML debugging examples</title>
    <!--[if lt IE 9]>
      <script src="//html5shiv.googlecode.com/svn/trunk/html5.js">
        <![endif]-->
    </head>
    <body data-gr-c-s-loaded="true">
      <nav class="dade-progress" style="background-color: #ccc; width: 100%; height: 10px; position: relative;">
        <div style="width: 100%; height: 100%; position: absolute; left: 0; top: 0; background-color: #000; opacity: 0.5; z-index: 1;></div>
        <div style="width: 100%; height: 100%; position: absolute; left: 0; top: 0; background-color: #fff; border-radius: 5px; z-index: 2;></div>
      </nav>
      <h1>HTML debugging examples</h1>
      <p>This page illustrates common HTML debugging examples. It includes sections on unclosed elements and badly nested elements, along with a live example of each. The developer tools are also used to inspect the page's DOM and CSS styles. The browser's developer tools menu is open, showing options like New Tab, History, and Developer Tools.
```



Browser developer tools

HTML debugging examples

p | 405.5 x 18.5

What causes errors in HTML?

- Unclosed elements: If an element is **not closed properly, then its effect can spread to areas you didn't intend**
- Badly nested elements: Nesting elements properly is also very important for code behaving correctly. **strong *strong emphasised? what is this?***
- Unclosed attributes: Another common source of HTML problems. Let's look at an example:

localhost:8000/debugging-example.html

Elements Console Sources Network

```
<!doctype html>
<html class="gr__localhost">
  <head>
    <meta charset="utf-8">
    <title>HTML debugging examples</title>
    <!--[if lt IE 9]>
      <script>
        src="//html5shiv.googlecode.com/svn/trunk/html5.js"
      </script>
    <![endif]-->
  </head>
  <body data-gr-c-s-loaded="true">
    <nav class="page-progress" style="background: rgba(240, 255, 255, 0.8); width: 100%; height: 0px; position: fixed; left: 0px; top: 0px; z-index: 9999;">
      <section class="read" style="width: 0%; height: inherit;">
      </section>
    </nav>
    <h1>HTML debugging examples</h1>
    <p>What causes errors in HTML?</p>
    </p> == $0
    <ul></ul>
  </body>
</html>
```

html.gr__localhost body p

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hover .cls +

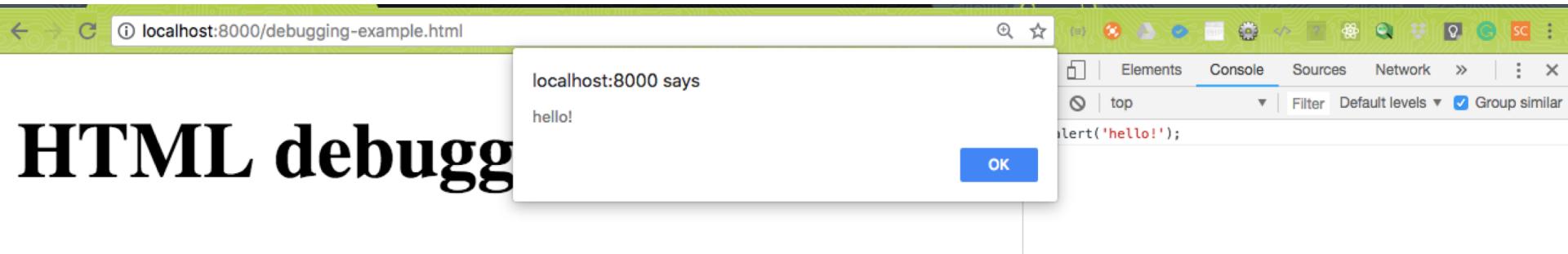
Console

top Filter Default levels Group similar



Using browser developer tool

- Using Console to type JavaScript code



HTML Validation

- The best strategy is to start by running your HTML page through the <https://validator.w3.org/>
 - Created and maintained by the W3C, the organization that looks after the specifications that define HTML, CSS, and other web technologies
- This webpage takes an HTML document as an input, goes through it, and gives you a report to tell you what is wrong with your HTML.



Sample HTML Validation

Secure | <https://validator.w3.org/nu/?doc=https%3A%2F%2Fkrunapon.github.io%2Fsections.html>

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://krunapon.github.io/sections.html>

Checker Input

Show source outline image report Options...

Check by address

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

1. **Error** Bad value `https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300|Sonsie+One` for attribute `href` on element `link`: Illegal character in query: `|` is not allowed.
From line 7, column 5; to line 7, column 126
`title>
 <link href="https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300|Sonsie+One" rel="stylesheet"
type="text/css"/>`

2. **Error** No `[p]` element in scope but a `[p]` end tag seen.
From line 52, column 68; to line 52, column 71
`e after hr</p>`



Agenda

- Hyperlinks
- Documents and website structure
- Debugging HTML
- **Multimedia and embedding**
- HTML tables
- HTML forms



Annotating images with figures and figure captions

- We can use the HTML5 `<figure>` and `<figcaption>` elements to provide a semantic container for figures, and clearly link the figure to the caption



[Khon Kaen University](#)



Audio and video on the web

- HTML5 have <video> and <audio> elements

```
1 | <video src="rabbit320.webm" controls>
2 |   <p>Your browser doesn't support HTML5 video. Here is a <a href="rabbit320.webm">link to
3 | </video>
```

- Attributes of <video> or <audio>
 - src: The src (source) attribute contains a path to the video you want to embed
 - controls: Users must be able to control video and audio playback



Sample usage of <video>

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Simple video example</title>
  </head>
  <body>
    <h1>Below is a simple video example</h1>
    <video src="rabbit320.webm" controls>
      <p>Your browser doesn't support HTML5 video. Here is a <a href="rabbit320.webm">link to the video</a> instead.</p>
    </video>
  </body>
</html>
```



Sample usage of <audio>

```
<audio controls>
```

```
  <source src="audios/viper.mp3" type="audio/mp3">
```

```
  <source src="audios/viper.ogg" type="audio/ogg">
```

```
    <p>Your browser doesn't support HTML5 audio. Here  
is a <a href="audios/viper.mp3">link to the audio</a>  
instead.</p>
```

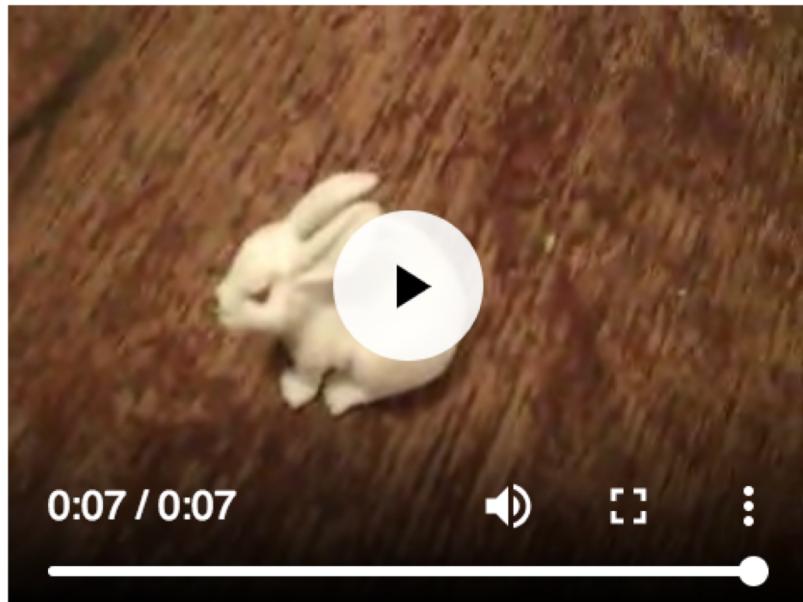
```
</audio>
```



Sample media files on html



Below is a simple video example



Agenda

- Hyperlinks
- Documents and website structure
- Debugging HTML
- Multimedia and embedding
- **HTML tables**
- HTML forms



HTML Tables

```
11 <table>
12   <tr>
13     <td>Hi, I'm your first cell.</td>
14     <td>I'm your second cell.</td>
15     <td>I'm your third cell.</td>
16     <td>I'm your fourth cell.</td>
17   </tr>
18   <tr>
19     <td>Second row, first cell.</td>
20     <td>Cell 2.</td>
21     <td>Cell 3.</td>
22     <td>Cell 4.</td>
23   </tr>
24 </table>
```

Hi, I'm your first cell.	I'm your second cell.	I'm your third cell.	I'm your fourth cell.
Second row, first cell.	Cell 2.	Cell 3.	Cell 4.

Adding headers with <th> elements

```
1 <table>
2   <tr>
3     <td>&nbsp;</td>
4     <td>Knocky</td>
5     <td>Flor</td>
6     <td>Ella</td>
7     <td>Juan</td>
8   </tr>
9   <tr>
10    <td>Breed</td>
11    <td>Jack Russell</td>
12    <td>Poodle</td>
13    <td>Streetdog</td>
14    <td>Cocker Spaniel</td>
15  </tr>
16  <tr>
17    <td>Age</td>
18    <td>16</td>
19    <td>9</td>
20    <td>10</td>
21    <td>5</td>
22  </tr>
```

	Knocky	Flor	Ella	Juan
Breed	Jack Russell	Poodle	Streetdog	Cocker Spaniel
Age	16	9	10	5



Allowing cells to span multiple rows and columns

```
<table>
  <tr>
    <th>Animals</th>
  </tr>
  <tr>
    <th>Hippopotamus</th>
  </tr>
  <tr>
    <th>Horse</th>
    <td>Mare</td>
  </tr>
  <tr>
    <td>Stallion</td>
  </tr>
```

Animals
Hippopotamus
Horse
Stallion
Mare



Example timetable.html



School timetable

	Mon	Tues	Wed	Thurs	Fri	Sat	Sun
1st period	English			German	Dutch		
2nd period	English	English		German	Dutch		
3rd period		German		German	Dutch		
4th period		English		English	Dutch		



File timetable.html (1/3)

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>School timetable</title>
6  <style>
7      html {
8          font-family: sans-serif;
9      }
10     table {
11         border-collapse: collapse;
12         border: 2px solid rgb(200,200,200);
13         letter-spacing: 1px;
14         font-size: 0.8rem;
15     }
16     td, th {
17         border: 1px solid rgb(190,190,190);
18         padding: 10px 20px;
19     }
20     td {
21         text-align: center;
22     }
23     caption {
24         padding: 10px;
25     }
26     </style>
27 </head>
28 <body>
29     <h1>School timetable</h1>
```



File timetable.html (2/3)

```
30
31    <table>
32    <tr>
33        <td>&nbsp;</td>
34        <th>Mon</th>
35        <th>Tues</th>
36        <th>Wed</th>
37        <th>Thurs</th>
38        <th>Fri</th>
39        <th>Sat</th>
40        <th>Sun</th>
41    </tr>
42    <tr>
43        <th>1st period</th>
44        <td>English</td>
45        <td>&nbsp;</td>
46        <td>&nbsp;</td>
47        <td>German</td>
48        <td>Dutch</td>
49        <td>&nbsp;</td>
50        <td>&nbsp;</td>
51    </tr>
52    <tr>
53        <th>2nd period</th>
54        <td>English</td>
55        <td>English</td>
56        <td>&nbsp;</td>
57        <td>German</td>
58        <td>Dutch</td>
59        <td>&nbsp;</td>
```



File timetable.html (3/3)

```
55      <td>English</td>
56      <td>&nbsp;</td>
57      <td>German</td>
58      <td>Dutch</td>
59      <td>&nbsp;</td>
60      <td>&nbsp;</td>
61      </tr>
62 ▼    <tr>
63      <th>3rd period</th>
64      <td>&nbsp;</td>
65      <td>German</td>
66      <td>&nbsp;</td>
67      <td>German</td>
68      <td>Dutch</td>
69      <td>&nbsp;</td>
70      <td>&nbsp;</td>
71      </tr>
72 ▼    <tr>
73      <th>4th period</th>
74      <td>&nbsp;</td>
75      <td>English</td>
76      <td>&nbsp;</td>
77      <td>English</td>
78      <td>Dutch</td>
79      <td>&nbsp;</td>
80      <td>&nbsp;</td>
81      </tr>
82      </table>
83  </body>
84  </html>
```



Exercise: KKU Presidential Candidates

KKU Presidential Candidates 2018

1.	กุลธิดา ท้วมสุข	คณะมนุษยศาสตร์และสังคม
2.	ชาญชัย พานทองวิริยะกุล	คณะแพทยศาสตร์
3.	บวรศิลป์ เชาว์ชื่น	คณะแพทยศาสตร์
4.	ศุภชัย ปทุมนาภุล	คณะวิศวกรรมศาสตร์
5.	อภิรักษ์ ศิริธรรมราชวัตร	คณะวิศวกรรมศาสตร์



Providing common styling to columns

- HTML has a method of defining styling information for an entire column of data all in one place
 - <col> and <colgroup> elements
- Example

Data 1	Data 2
Calcutta	Orange
Robots	Jazz



The First Solution

```
1 <table>
2   <tr>
3     <th>Data 1</th>
4     <th style="background-color: yellow">Data 2</th>
5   </tr>
6   <tr>
7     <td>Calcutta</td>
8     <td style="background-color: yellow">Orange</td>
9   </tr>
10  <tr>
11    <td>Robots</td>
12    <td style="background-color: yellow">Jazz</td>
13  </tr>
14 </table>
```

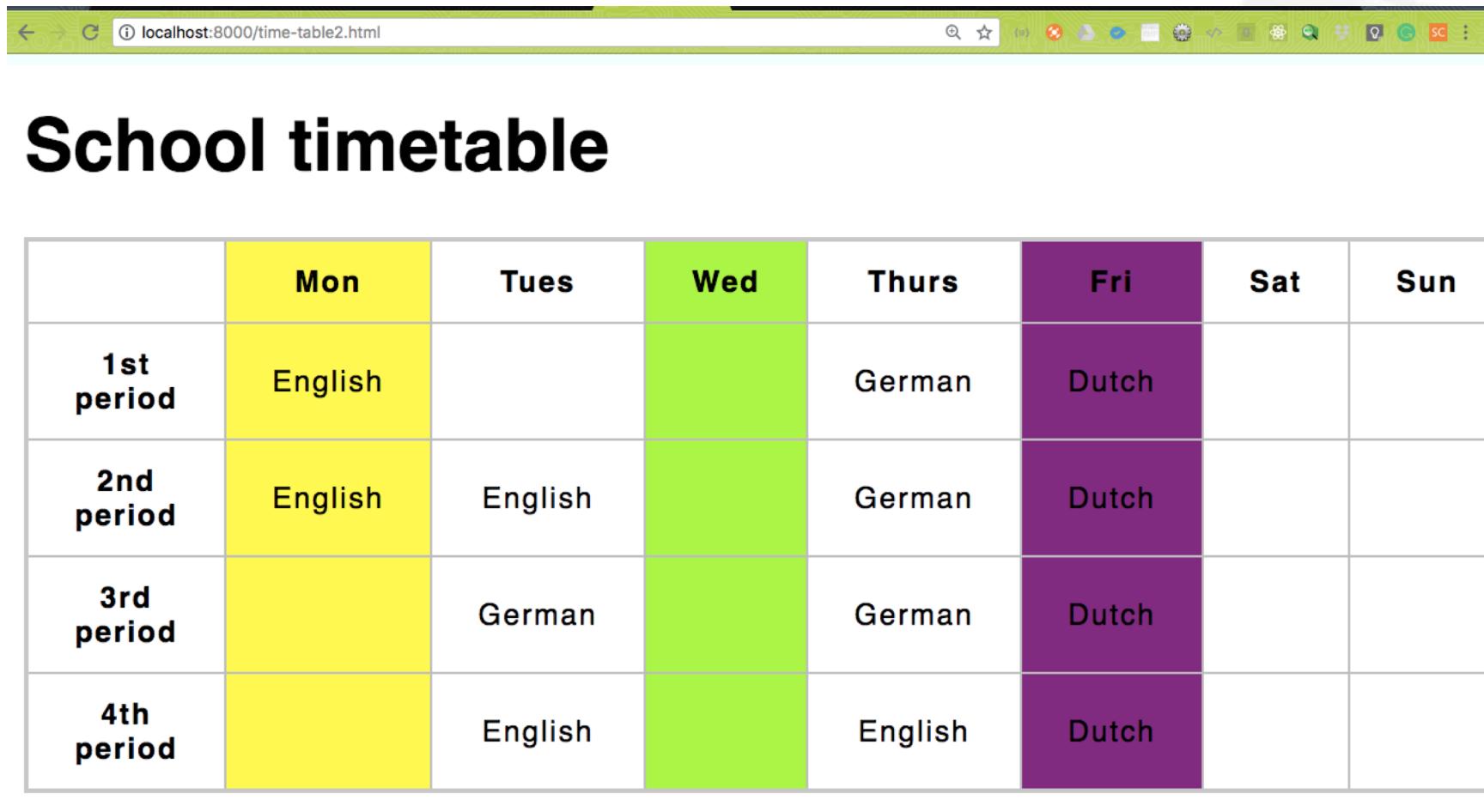


The Second Solution

```
1 <table>
2   <colgroup>
3     <col>
4     <col style="background-color: yellow">
5   </colgroup>
6   <tr>
7     <th>Data 1</th>
8     <th>Data 2</th>
9   </tr>
10  <tr>
11    <td>Calcutta</td>
12    <td>Orange</td>
13  </tr>
14  <tr>
15    <td>Robots</td>
16    <td>Jazz</td>
17  </tr>
18 </table>
```



Sample School Timetable



A screenshot of a web browser displaying a school timetable. The title "School timetable" is visible at the top of the page. The timetable is a grid with 8 columns representing the days of the week: Mon, Tues, Wed, Thurs, Fri, Sat, Sun, and an additional column for periods. The rows represent four periods. The days of the week are color-coded: Mon (yellow), Tues (light gray), Wed (green), Thurs (purple), Fri (dark purple), Sat (light gray), and Sun (light gray). The period columns are also color-coded: Period 1 (yellow), Period 2 (yellow), Period 3 (yellow), and Period 4 (yellow). The subjects listed are English, German, and Dutch.

	Mon	Tues	Wed	Thurs	Fri	Sat	Sun
1st period	English			German	Dutch		
2nd period	English	English		German	Dutch		
3rd period		German		German	Dutch		
4th period		English		English	Dutch		



Agenda

- Hyperlinks
- Documents and website structure
- Debugging HTML
- Multimedia and embedding
- HTML tables
- **HTML forms**



What are HTML forms?

- HTML Forms are one of the main points of interaction between a user and a web site or application
- They allow users to send data to the web site
- Most of the time that data is sent to the web server, but the web page can also intercept it to use it on its own.



Designing your form

- Three main uses of forms

	Commerce	Community	Productivity
User Objective	Obtain Info / Buy	Join Community	Get Things Done
Business Objective	Maximize Sales	Grow & Increase Engagement in the Community	Increase Content & Time Spent on Site
Example of Type of Site	E-Commerce	Social Networks	Online Banking
Example of a Form	Checkout Form	Registration Forms	Transfer of Funds



The six components of web forms (1/2)

- **Labels:** tell users what the corresponding input fields mean
- **Input fields:** enable users to provide feedback
 - Examples: text fields, password fields, check boxes, radio buttons, sliders and more
- **Actions:** there are links or buttons that, when pressed by the user, perform an action, such as submitting the form



The six components of web forms (2/2)

- **Help**
 - This provides assistance on how to fill out the form
- **Messages**
 - Messages give feedback to the user based on their input. They can be positive or negative
- **Validation**
 - These measures ensure that the data submitted by the user conforms to acceptable parameters



Sample Form Components

5. Message ➡ Please review information that you have entered.

1. Label ➡ First name*	2. Input Field ➡ Justin	3. Label ➡ Last name*	4. Input Field ➡ Mifsud
Your email address*		Repeat email*	
badEmail			
6. Validation ➡ Email address not valid	Note: no-one can see your email address.		

By email

saved 

Can't read the text in the box?    4. Help

Type the text above here*

Yes, I have read and I accept the [Skype Terms of Use](#) and the [Skype Privacy Statement](#)

I agree - Continue 3. Action

Skype's registration form contains all six components.



Tacking usability via three aspects of forms

- For a form to be usable, all three aspects need to be tackled
 - Relationship
 - Conversation
 - Appearance

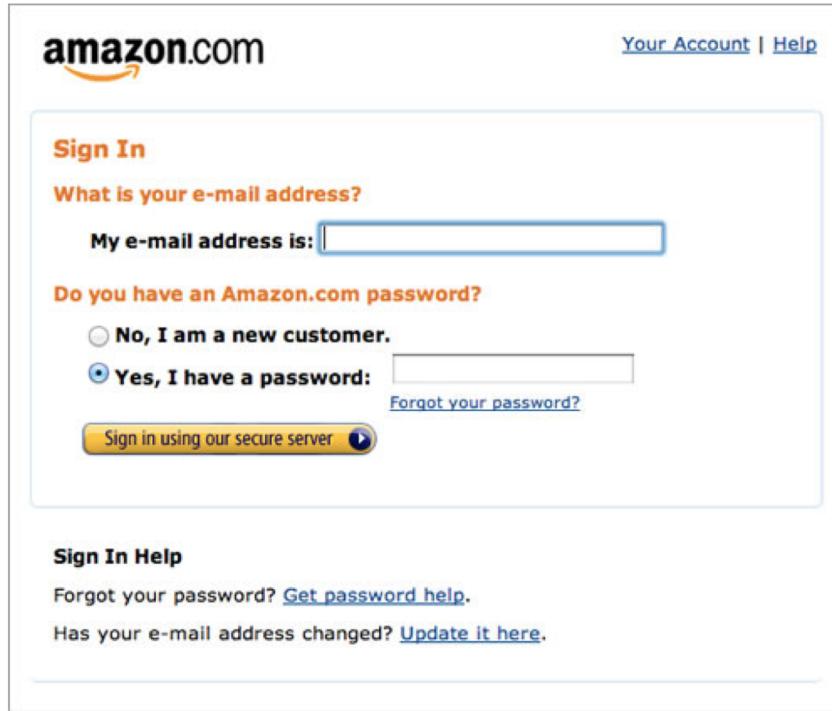


Aspect 1: relationship

- Every relationship has a goal
- Base the name of the form on its purpose
- Choose appropriate language and remove superfluous text
- Do not ask questions beyond the scope of the form
- Sudden changes in behavior or appearance



Sample sign in form



The screenshot shows the Amazon.com sign-in page. At the top left is the Amazon logo, and at the top right are links for "Your Account" and "Help". The main heading is "Sign In". Below it, the question "What is your e-mail address?" is displayed, followed by a text input field labeled "My e-mail address is: [REDACTED]". The next section asks "Do you have an Amazon.com password?", with two radio button options: "No, I am a new customer." (unchecked) and "Yes, I have a password:" (checked). To the right of the checked radio button is a password input field and a link "Forgot your password?". Below these fields is a yellow "Sign in using our secure server" button with a lock icon. At the bottom of the main form area, there is a "Sign In Help" section containing links for "Forgot your password? [Get password help.](#)" and "Has your e-mail address changed? [Update it here.](#)".

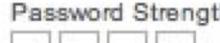
- ☞ Amazon, on the other hand, simplifies the process for registered and new customers.



Aspect 2: the conversation (1/2)

- A form is a conversation, not an interrogation
- Order the labels logically
 - Name should come before phone numbers
 - More involved questions should come towards at the end of the form
- Group related information
 - The flow from one set of questions to the next will better resemble a conversation

Sample form that groups related information

Select an ID and password	
Yahoo! ID and Email	<input type="text"/> @ <input type="text"/> .yahoo.com <input type="button" value="Check"/>
Password	<input type="password"/> Password Strength 
Re-type Password	<input type="password"/>
In case you forget your ID or password...	
Alternate Email (optional)	<input type="text"/>
Secret Question 1	<input type="button" value="– Select One –"/>
Your Answer	<input type="text"/>
Secret Question 2	<input type="button" value="– Select One –"/>
Your Answer	<input type="text"/>



Aspect 2: the conversation (2/2)

- Each label should **address one topic at a time**
- The **natural pauses** in a conversation will indicate where to introduce white space, how to group labels and whether to break the form up over multiple pages
- In any conversation, people get distracted by background noise. So, **remove clutter** such as banners and unnecessary navigation that might distract users from filling out the form.



Sample good registration form

The screenshot shows a registration form for creating a Dropbox account. At the top, there are two buttons: "Log in" (with a key icon) and "Create an Account" (with a pencil icon). Below these buttons, the title "Create a Dropbox Account" is centered. The form consists of four input fields: "First name" (text box), "Last name" (text box), "Email" (text box), and "Password" (text box). Below the password field is a checkbox labeled "I agree to Dropbox Terms of Service". A large blue "Create account" button is positioned below the checkbox. At the bottom of the form, there is a link "New to Dropbox? Check out our tour.".

Create a Dropbox Account

First name

Last name

Email

Password

I agree to [Dropbox Terms of Service](#)

New to Dropbox? [Check out our tour.](#)



Aspect 3: the appearance

- Labels: Individual words vs. sentences
 - If the purpose of a label is simple to understand, such as to ask for a name or telephone number, then a word or two should suffice
 - But a phrase or sentence might be necessary to eliminate ambiguity

The image shows the Amazon.com registration form. At the top left is the Amazon logo and at the top right are links for "Your Account" and "Help". The main title "Registration" is in red, followed by the sub-instruction "New to Amazon.com? Register Below.". There are four input fields: "My name is:", "My e-mail address is:", "Type it again:", and "My mobile phone number is: (Optional)". Below these is a section titled "Protect your information with a password" with the sub-instruction "This will be your only Amazon.com password." It contains two more input fields: "Enter a new password:" and "Type it again:". A "Create account" button is at the bottom right. At the very bottom of the form, there are links for "Conditions of Use" and "Privacy Notice" with the copyright notice "© 1996-2011, Amazon.com, Inc. or its affiliates".

Amazon's registration form contains full sentences, whereas individual words would have sufficed.

Aspect 3: the appearance

- **Sentence case vs. title case**
 - Should it be “Name and Surname” or “Name and surname - Sentence case is slightly easier — and thus faster — to follow grammatically than title case
 - One thing is for sure: never use all caps, or else the form would look unprofessional and be difficult to scan.



Unpreferable Registration Form (1/2)

Sign In or Create an Account

To use this feature, you must sign in or create an account.

If you already have an account, please sign in.

EMAIL ADDRESS: *

RE-ENTER EMAIL ADDRESS: *

FIRST NAME: *

LAST NAME: *

PASSWORD: *

CONFIRM PASSWORD: *

(Use 6-15 letters, numbers, or numeric symbols. CaSe SeNsItIvE. No spaces.)

SECURITY QUESTION *

- ⌚ See how difficult it is to quickly scan the labels in Barnes & Noble's registration form?



Unpreferable Registration Form (2/2)

The screenshot shows a registration form with a pink header navigation bar containing links for BRUSHES, EYES, LIPS, FACE, PALETTES, KITS, NEW ITEMS, and CLEARANCE. Below the header, there are two columns: 'PERSONAL INFORMATION' on the left and 'LOGIN INFORMATION' on the right. The 'PERSONAL INFORMATION' section includes fields for FIRST NAME*, LAST NAME*, and EMAIL ADDRESS*. The 'LOGIN INFORMATION' section includes fields for PASSWORD* and CONFIRM PASSWORD*. At the bottom is a large red 'SUBMIT' button.

- Forms should never consist of more than one column. Notice how easy it is to ignore the column on the right here on Makeup Geek (not to mention the note about “Required fields” at the bottom).



Aspect 3: the appearance

- **Input fields**
 - **Customizing input fields:** simple is often the most useful. Keep input fields as close to their unaltered HTML rendering as possible
 - **Restricting the format of input fields**
 - Instead of displaying MM/DD/YYYY next to a text field for a date, consider using three drop-down fields or, better yet, a calendar control
 - **Mandatory vs. optional fields**
 - Clearly distinguish which input fields cannot be left blank by the user
 - The convention is to use an asterisk (*).



Sample form? Preferable or unpreferable

Contact Us Form
Fields marked with * are mandatory

Name	*
E-mail	*
Mobile/Phone	
Service Required	– Select –
Comments	*



Aspect 3: the appearance

- **Actions**
 - Primary vs. secondary actions
 - Primary actions are links and buttons in a form that performs essential “final” functionality, such as “Save” and “Submit”
 - Secondary actions, such as “Back” and “Cancel”, enable users to retract data that they have entered

Primary vs. secondary actions

Before you submit this form, be certain that you have entered all of the required information. If you fail to submit a required field you will receive an error message. If you receive an error, please use the back button on your browser to return to this form and try again.

The following fields are required before you may successfully submit this form:

- Student number
- Last name
- First name
- Email Address
- Address
- City, State and Zip
- Home Phone
- Course Registration
- Total Cost
- Payment information
- Additional information

- ☞ Not clearly distinguishing between primary and secondary actions can easily lead to failure. The above action buttons are found at the end of a lengthy form for enrolling in St. Louis Community College. Just imagine pressing the “Reset Form” button by accident.



Naming conversions

- Avoid generic words such as “Submit” for actions, because they give the impression that the form itself is generic
- Descriptive words and phrases, such as “Join LinkedIn,” are preferred.



Example of unpreferable form

Coke Music Profile Registration

Fill out the form below to create an account.

* indicates required fields

* Email Address:

* Date of birth: Month | Date | Year

* I live in US


Can't read this?

* Enter Security Code:

Security Code is case-sensitive

CANCEL **SUBMIT**

- Although Coca-Cola correctly gives more importance to the primary action button, it settles for the generic word "Submit." "Register with us" would have been more helpful.



Aspect 3: The Appearance >> Help

- **User-triggered and dynamic help**
 - Rather than include help text next to each input field, show it only where required
 - You could show an icon next to an input field that the user can click on when they need help for that field
 - Even better, show help dynamically when the user clicks into an input field to enter data

Sample dynamic help message

The screenshot shows a registration form for Skype. At the top, there's a question "How do you intend to use Skype?" with a dropdown menu labeled "Select". Below it, there's a field for "Skype Name*" containing the text "justin". A message below the field says "Skype Name not available". Underneath, there's a "Suggestions" section with three radio buttons: "justin52114", "justin91859", and "justin45950". At the bottom of the form, a note says "Note: only choose a name you have a right to use..". To the right of the "justin" input field, a blue callout box provides dynamic help for "What is my Skype Name?". It explains that a Skype Name is a unique username for Skype, must be between 6-32 characters, start with a letter, and contain only letters and numbers.

- Skype's registration form contains both user-triggered help (the blue box that is triggered by clicking the question mark) and dynamic help (the suggested user names).



Aspect 3: The Appearance >> Messages

- **Smart defaults**
 - Use smart defaults to make the user's completion of the form faster and more accurate
 - For example, pre-select the user's country based on their IP address
 - But use these with caution, because users tend to leave pre-selected fields as they are.

Example of smart defaults

The screenshot shows the Twitter registration form. The fields and their validation status are:

- Name: Justin Mifsud (✓ Name looks great.)
- Email: justinmifsud@gmail.com (✗ This email is already registered. Want to [login](#) or [recover your password?](#))
- Password: (✓ Password is perfect!)
- Username: justinmifsud (✗ This username is already taken!)

Suggestions: MifsudJustin · MifsudJustin · justin_mifsud

Keep me logged-in on this computer.

By clicking the button, you agree to the terms below:
These Terms of Service ("Terms") govern your access to and use of the services and Twitter's websites (the "Services"), and any information, text, graphics, photos or other materials

Printable versions:
[Terms of Service](#) · [Privacy Policy](#)

Create my account

- ⌚ Twitter's registration form uses both dynamic validation (for the name, email address, password and user name) and smart defaults ("Keep me logged in").



Let's build a simple contact form

A hand-drawn sketch of a simple contact form on grid paper. The form is enclosed in a large rectangular border. At the top left, the word "CONTACT" is written above a horizontal line. Below this, there are three input fields: "Name:" followed by a long rectangular box, "E-mail:" followed by another long rectangular box, and "Message:" followed by a large rectangular box for a multi-line message. At the bottom center, there is a rounded rectangular button with the text "SEND YOUR MESSAGE" inside.



Sample code contact form

```
1 <form action="/my-handling-form-page" method="post">
2   <div>
3     <label for="name">Name:</label>
4     <input type="text" id="name" name="user_name">
5   </div>
6   <div>
7     <label for="mail">E-mail:</label>
8     <input type="email" id="mail" name="user_mail">
9   </div>
10  <div>
11    <label for="msg">Message:</label>
12    <textarea id="msg" name="user_message"></textarea>
13  </div>
14 </form>
```



The <form> element

- All HTML forms start with a <form> element like this

```
1 | <form action="/my-handling-form-page" method="post">  
2 |  
3 | </form>
```



The <action> and <method> attributes

- The action attribute defines the location (URL) where the form's collected data should be sent when it is submitted.
- The method attribute defines which HTTP method to send the data with (it can be "get" or "post").



The <input> element

- On the <input> element, the most important attribute is the type attribute
 - We use the value **text** for the first input – the default value for this attribute
 - It represents a basic single-line text field that accepts any kind of text input
 - We use the value **email** that defines a single-line text field that only accepts a well-formed email address

The <button> element

```
1 | <div class="button">  
2 |   <button type="submit">Send your message</button>  
3 | </div>
```

- A click on a **submit** button (the default value) sends the form's data to the web page defined by the **action** attribute of the <form> element
- A click on a **reset** button resets all the form widgets to their default value immediately



The First Form

← → C ⓘ localhost:8000/form1.html

Name:

E-mail:

Message:

Send your message



The Second Form with <style>

A screenshot of a web browser window displaying a form. The address bar shows the URL `localhost:8000/form1.html`. The form itself has a light gray background and rounded corners. It contains three text input fields labeled "Name:", "E-mail:", and "Message:". Below the "Message:" field is a large empty text area. At the bottom right of the form is a button labeled "Send your message".

Name:

E-mail:

Message:



Adding style to the form (1/3)

- First of all, add a `<style>` element to your page, inside your HTML head
- Inside the style tags, add the following CSS

https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Your_first_HTML_form

```
1  form {  
2      /* Just to center the form on the page */  
3      margin: 0 auto;  
4      width: 400px;  
5      /* To see the outline of the form */  
6      padding: 1em;  
7      border: 1px solid #CCC;  
8      border-radius: 1em;  
9  }  
10  
11  form div + div {  
12      margin-top: 1em;  
13  }
```

Adding style to the form (2/3)

```
15 label {  
16     /* To make sure that all labels have the same size and are properly aligned */  
17     display: inline-block;  
18     width: 90px;  
19     text-align: right;  
20 }  
21  
22 input, textarea {  
23     /* To make sure that all text fields have the same font settings  
24         By default, textareas have a monospace font */  
25     font: 1em sans-serif;  
26  
27     /* To give the same size to all text fields */  
28     width: 300px;  
29     box-sizing: border-box;  
30  
31     /* To harmonize the look & feel of text field border */  
32     border: 1px solid #999;  
33 }
```



Adding style to the form (3/3)

```
34  
35 input:focus, textarea:focus {  
36     /* To give a little highlight on active elements */  
37     border-color: #000;  
38 }  
39  
40 textarea {  
41     /* To properly align multiline text fields with their labels */  
42     vertical-align: top;  
43  
44     /* To give enough room to type some text */  
45     height: 5em;  
46 }  
47  
48 .button {  
49     /* To position the buttons to the same position of the text fields */  
50     padding-left: 90px; /* same size as the label elements */  
51 }  
52  
53 button {  
54     /* This extra margin represent roughly the same space as the space  
55         between the labels and their text fields */  
56     margin-left: .5em;  
57 }
```



Sending form data to your web server

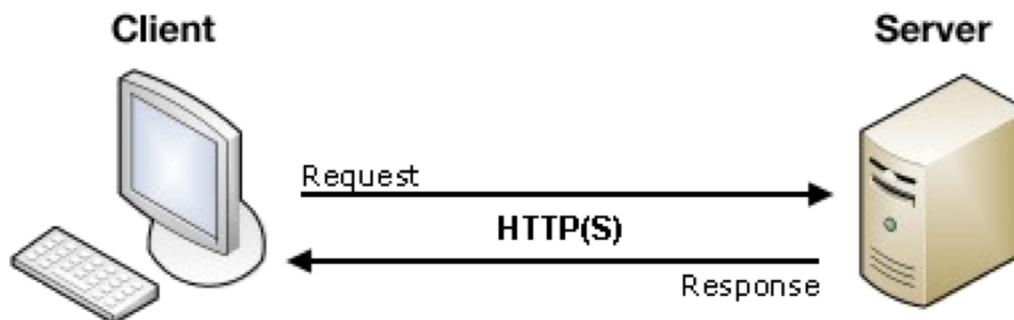
- The <form> element will define where and how to send the data through the action attribute and the method attribute
- We also need to give a name to our data

```
1 <form action="/my-handling-form-page" method="post">
2   <div>
3     <label for="name">Name:</label>
4     <input type="text" id="name" name="user_name" />
5   <div>
6     <div>
7       <label for="mail">E-mail:</label>
8       <input type="email" id="mail" name="user_email" />
9     </div>
```



Where does the data go?

- About client/server architecture
 - The web is based on a very basic client/server architecture that can be summarized as follows:
 - A client (usually a Web browser) sends a request to a server (most of the time a web server like Apache, Nginx, IIS, Tomcat, etc.) using the HTTP protocol



The action attribute

- This attribute defines where the data gets sent
- In this example, the data is sent to an absolute URL
 - <form action="<http://foo.com>">
- Here we use a relative URL – the data is sent to a different URL on the server
 - <form action="/somewhere_else">
- When specified with no attributes, as below, the <form> data is sent to the same page that the form is present on:
 - <form>



The method attribute

- This attribute defines how data is sent
- The GET method
 - In this case, the browser sends an empty body
 - Because the body is empty, if a form is sent using this method the data sent to the server is appended to the URL
- The POST method
 - If a form is sent using this method, you get no data appended to the URL and the data is appended to the body of the HTTP request

Example sending the form data

```
1 <form action="http://foo.com" method="post">
2   <div>
3     <label for="say">What greeting do you want to say?</label>
4     <input name="say" id="say" value="Hi">
5   </div>
6   <div>
7     <label for="to">Who do you want to say it to?</label>
8     <input name="to" id="to" value="Mom">
9   </div>
10  <div>
11    <button>Send my greetings</button>
12  </div>
13 </form>
```



Example HTTP request data

```
1 POST / HTTP/1.1
2 Host: foo.com
3 Content-Type: application/x-www-form-urlencoded
4 Content-Length: 13
5
6 say=Hi&to=Mom
```



Viewing HTTP requests

- If you want to see HTTP requests, you need to use tools such as the [Firefox Network Monitor](#) or the [Chrome Developer Tools](#)

The screenshot shows the Network tab of the Chrome Developer Tools. At the top, there's a toolbar with tabs for Elements, Console, Sources, Network (which is selected), Timeline, Profiles, Application, Security, and Audits. Below the toolbar, there are several controls: a red circle, a stop sign, a play/pause button, and dropdown menus for View, Preserve log, Disable cache, Offline, and No throttling. A 'Filter' input field with a Regex checkbox and a 'All' tab are also present. The main area displays a timeline from 500ms to 6000ms with various colored bars representing request durations. Below the timeline, a table lists network requests. The first row shows 'Name' and 'Headers' columns. The 'Name' column lists items like 'foo.com', 'www.foo.com', 'load_style.css', and 'cordovaimage.jpg'. The 'Headers' column shows details for the 'foo.com' request, including 'Upgrade-Insecure-Requests: 1', 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36', and a 'Form Data' section with 'say: Hi' and 'to: Mom'. At the bottom left, it says '23 requests | 22.0KB transferred | ...'.



On the server side: retrieving the data

- PHP offers some global objects to access the data
 - Assuming you've used the POST method, the following example just takes the data and displays it to the user

```
1 <?php
2 // The global $_POST variable allows you to access the data sent with the POST method by name
3 // To access the data sent with the GET method, you can use $_GET
4 $say = htmlspecialchars($_POST['say']);
5 $to = htmlspecialchars($_POST['to']);
6
7 echo $say, ' ', $to;
8 ?>
```



Other languages and frameworks

- There are many other server-side technologies you can use for form handling, including Python, Perl, Java, .Net, Ruby
- It's more common to use of the many nice frameworks that make handling forms easier
 - [Symfony](#) for PHP
 - [Django](#) for Python (a bit more heavyweight than [Flask](#), but with more tools and options).
 - [Express](#) for Node.js
 - [Ruby On Rails](#) for Ruby
 - [Grails](#) for Java



Calling server code from your form

- Change the method to GET
- Try to call the server code at

<https://fb.kku.ac.th/krunapon/html/handle-form1.php>

A screenshot of a web browser window. The address bar shows the URL `localhost:8000/form1.html`. The page contains a form with three text input fields and one button. The first field is labeled "Name:" with the value "kanda". The second field is labeled "E-mail:" with the value "kanda.runapongsa@gmail.com". The third field is labeled "Message:" with the value "Hello". Below the fields is a button labeled "Send your message".

Secure | https://fb.kku.ac.th/krunapon/html/handle-form1.php?user_name=kanda&user_email=kanda.runapongsa%40gmail.com...

kanda has email as **kanda.runapongsa@gmail.com** with the message **Hello**

References

- <https://developer.mozilla.org/en-US/docs/Learn/HTML>
- [Extensive Guide To Web Form Usability](#)

