Comp 125 - Visual Information Processing

Spring Semester 2018 - week 5 - monday

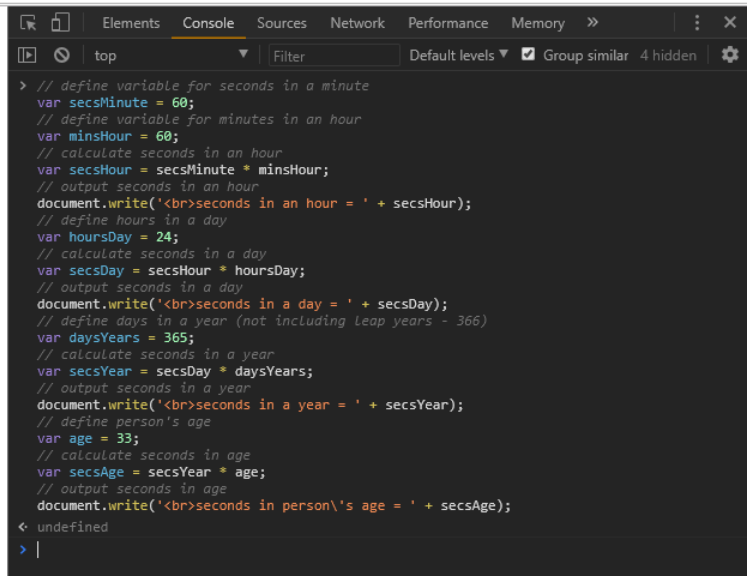Dr Nick Hayward

Fun exercise - using variables and operators

- calculate the **number of seconds in an hour**
- using the **number of seconds in an hour**, calculate the **number of seconds in a day**
- using **number of seconds in a day**, calculate the **number of seconds in a year**
- using **number of seconds in a year**, calculate the **number of seconds in your current age** in years, e.g. 22 years

Output each answer to the document with a line break between each result.

Fun exercise - using variables and operators

one possible answer using current notes...


```
seconds in an hour = 3600
seconds in a day = 86400
seconds in a year = 31536000
seconds in person's age = 1040688000
```

```
Elements    Console    Sources    Network    Performance    Memory    »

top                     ▼  Filter                Default levels ▼  ☑ Group similar  4 hidden

> // define variable for seconds in a minute
  var secsMinute = 60;
  // define variable for minutes in an hour
  var minsHour = 60;
  // calculate seconds in an hour
  var secsHour = secsMinute * minsHour;
  // output seconds in an hour
  document.write('<br>seconds in an hour = ' + secsHour);
  // define hours in a day
  var hoursDay = 24;
  // calculate seconds in a day
  var secsDay = secsHour * hoursDay;
  // output seconds in a day
  document.write('<br>seconds in a day = ' + secsDay);
  // define days in a year (not including leap years - 366)
  var daysYears = 365;
  // calculate seconds in a year
  var secsYear = secsDay * daysYears;
  // output seconds in a year
  document.write('<br>seconds in a year = ' + secsYear);
  // define person's age
  var age = 33;
  // calculate seconds in age
  var secsAge = secsYear * age;
  // output seconds in age
  document.write('<br>seconds in person\'s age = ' + secsAge);
< undefined
> |
```

Fun Exercise - using variables and operators

HTML Basics - <body> - part I

intro

- to define the main body of the web page we use the <body> element
- headings can be created using variants of
  - *<h1>, <h2>.....<h6>*
- we can now add some simple text in a <p> element

```
<p>...</p>
```

- add a line break using the <br> element
  - *
 for strict XHTML void*

- <hr> element adds a horizontal line
  - *<hr /> for strict XHTML void*
  - *implies rendering division*
  - *instead of defined structural divide...*

- comments can also be added through our HTML

```
<!-- comment... -->
```

HTML Basics - <body> - part 2

linking

- linking is an inevitable part of web design and HTML usage

- can be considered within three different contexts
  - *linking to an external site*
  - *linking to another page within the same site*
  - *linking different parts of the same page*

- add links to text and images within the HTML

- <a> element for links plus required attributes, e.g.

```html
<!-- external link -->
<a href="http://www.google.com/">Google</a>
<!-- email link -->
<a href="mailto:name@email.com">Email</a>
<!-- internal page link -->
<a href="another_page.html">another page</a>
<!-- define internal anchor - using name attribute -->
<a name="anchor">Internal anchor</a>
<!-- define internal anchor - using ID attribute -->
<a id="anchor">Anchor</a>
<!-- internal anchor link -->
<a href="#anchor">Visit internal anchor</a>
<!-- internal anchor link on another page -->
<a href="/another_page.html#anchor">Visit internal anchor</a>
<!-- internal anchor link on a page on an external site -->
<a href="https://www.test.com/test.html#anchor">Visit internal anchor on external site</a>
```

- Demo - HTML - Internal Anchor

HTML Basics - \<body> - part 3

linking - cont'd

- standard attributes supported by \<a> element include
  - *class, id, lang, style, title...*

- optional attributes are available for \<a> element including
  - *target, href, name...*

- target attribute specifies where the link will be opened relative to the current browser window

- possible attribute values include

```
<!-- open link in new window or tab -->
_blank
<!-- same frame -->
_self
<!-- open within parent frameset -->
_parent
<!-- open in the same window -->
_top
```

HTML Basics - <body> - part 4

images

- <img> allows us to embed an image within a web page
- <img> element requires a minimum *src* attribute

```
<img src="image.jpg" />
<img src="image.jpg">
```

- other optional attributes include
  - *class, id, alt, title, width, height...*
- use images as links
- image maps

```
<map name="textmap">
  <area shape="rect" coords="..." alt="Quote 1" href="notes1.html" />
</map>
```

HTML Basics - <body> - part 5

tables

- organise data within a table starting with the `<table>` element
- three primary child elements include
  - *table row, table header, table data*
  - *<tr>, <th>, <td>*

```
<table>
  <caption>424 - basic test table</caption>
  <tr>
    <th>heading 1</th>
    <th>heading 2</th>
  </tr>
  <tr>
    <td>row 1, cell 1</td>
    <td>row 2, cell 2</td>
  </tr>
</table>
```

- also add a `<caption>`
- span multiple columns using the `colspan` attribute
- span multiple rows using the `rowspan` attribute

lists

- unordered list <ul>, ordered list <ol>, definition list <dl>

- and <ol> contains list items

```
<ul>
   <li>...</li>
</ul>
```

```
<ol>
   <li></li>
</ol>
```

- definition list uses <dt> for the item, and <dd> for the definition

```
<dl>
   <dt>Game 1</dt>
   <dd>our definition</dd>
</dl>
```

HTML Basics - <body> - part 7

forms

- used to capture data input by a user, which can then be processed by the server
- `<form>` element acts as the parent wrapper for a form
- `<input>` element for user input includes options using the *type* attribute
  - *text, password, radio, checkbox, submit*

```html
<form>
  Text field: <input type="text" name="textfield" />
</form>
```

- process forms using
  - *e.g. JavaScript...*