Digital Innovation

Intelligent Interface of Human-Machine

Tutor: cch (diffusion)

```
Nothing is no more popular than HTML!
```

HTML

A popular format of data suit for any device and anybody!

Template

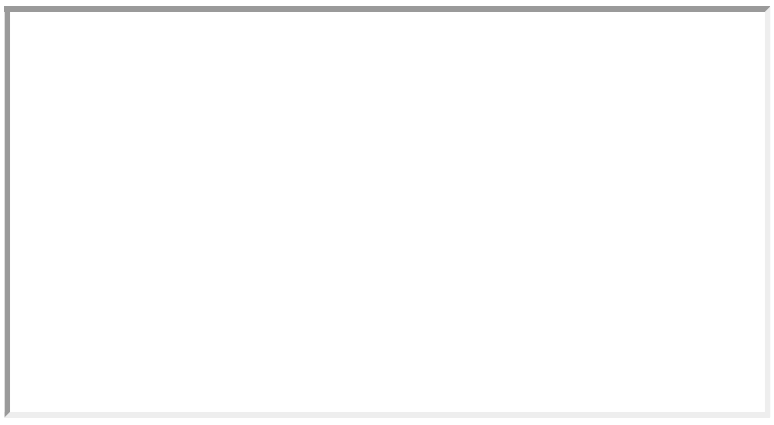
```
<html>
   <meta name="viewport" content="user-scalable=no, initial-scale=1, maxi
mum-scale=1, minimum-scale=1, width=device-width, height=device-height, t
arget-densitydpi=device-dpi" />
   <head>Page Title</head>
   <script type="text/javascript">...</script>
   <style type="text/css">...</style>
   <body>
      I'm here.
   </body>
</html>
```

- (verbatim mode) I'm here,
- I'm here, (by <big>... </big> tag)
- *I'm here*, (by <i>... <i> tag)
- I'm here, (by <font color="red">... </color> tag)

In [8]:

```
from IPython.display import HTML
HTML("<iframe src=1/here.html width=50% height=20%></iframe>")
```

Out[8]:



Code

```
<html>
   <meta name="viewport" content="user-scalable=no, initial-scale=1,
     maximum-scale=1, minimum-scale=1,
     width=device-width, height=device-height" />
   <body>
      I'm here.<br>
       <font color="red"> I'm here.</font><br>
       <code style="background-color:red;color:white;"> I'm here.</code><
br>
   </body>
</html>
```

Note

The *meta* tag with "viewport" used above is initially introduced for recent mobile devices device but is also popular in general HTML codes.

HTML Programming Environment

- HTML, CSS and Javascript
- Development Tool: Seamonkey,Gimp

Second Lecture (Interactivity)

Input a number, X, and calculate its square power, X^2:

Input: 6

Squre of input = **36**

Calculate

Main HTML Part

define a form, waiting for data input to work:

```
<form action="#">
  Input:
  <input id="real" type="numeric" name="real"
         min="0" max="100" step="1" value="6">
  <br>
     Squre of input = <b id="boldStuff">36</b>
  <br>
  <input value="Calculate"
     onclick="calcSquare(this.form.real.value)"
     type="button">
</form>
```

Javascript Function

Calculate the square of Input:

```
<script type="text/javascript">
    function calcSquare(real)
    {
      var result=real*real;
      document.getElementById('boldStuff').innerHTML = result;
    }
</script>
```

or

```
<script type="text/javascript">
    function calcSquare()
    {
      var realval=document.getElementById('real').value;
      var result=realval*realval;
      document.getElementById('boldStuff').innerHTML = result;
    }
</script>
```

Input: 6

Squre of input = **36**

Calculate

Note

- above result enhanced by some CSS

```
   <div style="font-family: Georgia, serif;background-color:gainsboro; bo
rder:solid black; width:300px; padding:20px;">
```

- same effect by standard CSS syntax

```
<style type="text/css">
   form,input {
         font-family:  Georgia, serif;
   }
</style>
```

Input: 6

Squre of input = **36**

Calculate

Code

```
<html>
   <meta name="viewport" content="user-scalable=no, initial-scale=1, maxi
mum-scale=1, minimum-scale=1, width=device-width, height=device-height" /
>
   <style type="text/css">
      form {
            font-family:  Georgia, serif;
            background-color:gainsboro;
            border:solid black;
            width:300px;
            padding:20px;
      }
      input {
            font-family:  Georgia, serif;
      }
   </style>

   <body>
      <form action="#">
     Input:
     <input id="real" type="numeric" name="real"
            min="0" max="100" step="1" value="6">
     <p />
        Squre of input = <b id="boldStuff">36</b>
        <br>
        <input value="Calculate"
           onclick="calcSquare(this.form.real.value)"
           type="button">
    </form>
    <script type="text/javascript">
         function calcSquare(real)
         {
           var result=real*real;
           document.getElementById('boldStuff').innerHTML = result;
         }
     </script>
    </body>
</html>
```

Computer Practice

- Complete the example file in HTML format.

Next Lecture

How to make an App with the HTML contents:

- Hello World (ipynb) (1/index.ipynb)
- Hello World (HTML) (1/index.html)

App's Internationalization (i18n)

1. Startup **Android Studio** (or **Eclipse**) and create a project, named "test";
2. Structure of projects, *src,res/[layout,values]*;
3. Setup "run configuration", create a new AVD, and test it;
4. copy whole the directory, res/values, and rename as "res/values-zh-rTW" for **Traditional Chinese Language**;
5. modify **string.xml** in the directory ceated by last step into Traditional Chinese;
6. change language of AVD to test the locale.
7. use mobile device as the test system.

In [2]:

```
%%bash

jupyter nbconvert Com-2016-1.ipynb
```

```
[NbConvertApp] Converting notebook Com-2016-1.ipynb to html
[NbConvertApp] Writing 285435 bytes to Com-2016-1.html
```

In [ ]: