- HTML forms are used to *collect user input*.
- The **\<form\>** element defines an HTML form.
- An HTML form contains *form elements* which can be grouped into the following tags :
  - `<input>`
    - `type=text|password|radio|checkbox|file|submit|reset|button|image|hidden`
  - `<select> and <option>`
  - `<textarea>`
  - `<button>`

common input fields : type attribute values (1)

- Example
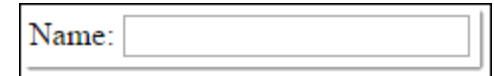 - See the code on next slide

common input fields : type attribute values (2)

- ## Examples

Name: `<input` **type**=`"`**text**`"` name=`"username" />`
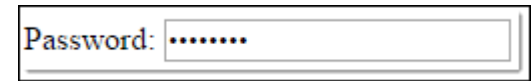

- **text** : define a single-line text field that a user can enter text into

Password: `<input` **type**=`"`**password**`"` name=`"password" />`

- **password** : define a password field (characters are *masked*)

Gender:

`<input` **type**=`"`**radio**`"` name=`"gender"` value=`"M"`

checked=`"checked" />` Male

`<input` **type**=`"`**radio**`"` name=`"gender"` value=`"F" />` Female

- **radio** : define radio buttons which let a user select only one of the choices
- cannot "*check*" one radio button from a group of radio buttons with the *same name*

common input fields : type attribute values (3)

- ## Examples

`<input `**`type`**`="`**`checkbox`**`" name="L1" value="Cantonese"` ☑ Cantonese ☐ English `checked="checked" /> Cantonese`

`<input `**`type`**`="`**`checkbox`**`" name="L2" value="English" /> English`

- **`checkbox`** : define checkboxes which let a user select one or more options
- Also, you can set an "array-like" name (`name="lang[]"`) for all checkboxes

`Submit a recent photo: <input `**`type`**`="`**`file`**`" name="photo" />`

- **`file`** : define a file-select field or a "Browse..." button for file upload
- Also need to set `method="post" enctype="multipart/form-data"` in the <form> element

Submit a recent photo: | Choose File | No file chosen

`<input `**`type`**`="`**`hidden`**`" name="city" value="Hong Kong" />`

- **`hidden`** : define a hidden field (not visible to a user) and the value can be changed by JavaScript code if necessary

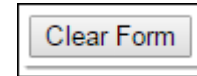# common input fields : type attribute values (4)

- ## Examples

```
<input type="submit" value="Send Form" />
```
Send Form

```
<button type="submit">Send Form</button>
```

- **submit** : define a submit button to send out the form data

```
<input type="reset" value="Clear Form" />
```
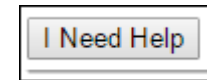Clear Form

```
<button type="reset">Clear Form</button>
```

- **reset** : define a reset button which can reset all form values to default values

```
<input type="button" value="I Need Help"
```
I Need Help
```
onclick="alert('Call hotline 22338899 for help');" />
<button type="button" onclick="alert('Call hotline 22338899
for help');">I Need Help</button>
```

- **button** : define a clickable button to trigger the execution of JavaScript code

```
<input type="image" name="save" src="images/icon.gif" />
```

- **image** : define an image as the submit button. Useful for *server-side* image map
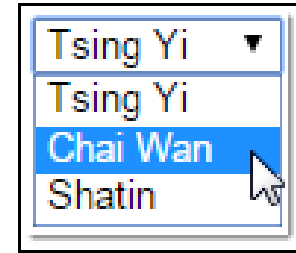
<form> tag attributes

- ## Examples

```
<input type="submit" value="Send Form" />
<form method="post" enctype="multipart/form-data"
action="read_data.php" name="register_form">
```

Attribute	Description
action	Specifies an address (url) where to submit the form.
enctype	Specifies the encoding of the submitted data (default: is url-encoded). Possible values are : • **application/x-www-form-urlencoded** (*default value* if the attribute is not specified) • **multipart/form-data** : for <input type="file"> elements
method	Specifies the HTTP method used when submitting the form. Possible values are : • **get** : form data are appended to the action attribute URI with a '?' as separator, and the resulting URI is sent to the server. (*default value* if the attribute is not specified) • **post** : form data are included in the body of the form and sent to the server.
name	Specifies a name used to identify the form (for retrieving form elements by JavaScript).

<select> element (Dropdown List)

- ## Example (single selection only)

```
<select name="campus">
  <option value="TY">Tsing Yi</option>
  <option value="CW">Chai Wan</option>
  <option value="ST">Shatin</option>
</select>
```

- ## Example (multiple selections)

```
<select name="speak" size="2" multiple="multiple">
  <option value="ENG">English</option>
  <option value="CTN">Cantonese</option>
  <option value="PTH">Putunghua</option>
</select>
```

- **Ctrl+Click** to select multiple options
- **multiple**="multiple" : Specifies that multiple options can be selected
- **size** : Defines the number of visible options in a dropdown list

<select> element (Dropdown List) – con't

- Example (group related options)

```
<select>
  <optgroup label="Fruit">
    <option>Apple</option>
    <option>Orange</option>
  </optgroup>
  <optgroup label="Vegetable">
    <option>Onion</option>
    <option>Tomato</option>
  </optgroup>
</select>
```
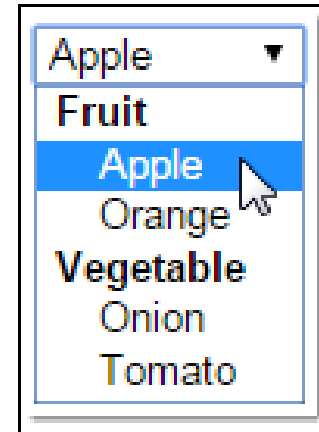
- **optgroup** : this tag is used to group related options
- If you have a long list of options, a user can easily handle groups of related options

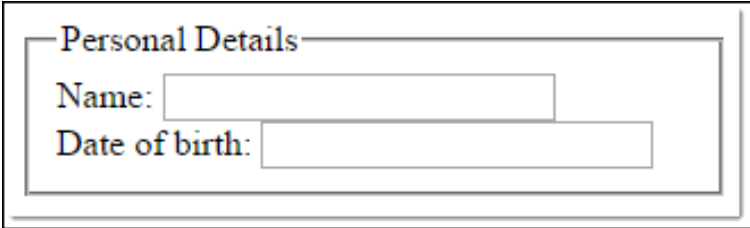# <fieldset> & <legend> element - Group Form Data

- Example

```
<fieldset>
   <legend>Personal Details</legend>
   Name: <input type="text" /><br />
   Date of birth: <input type="text" />
</fieldset>
```

- **fieldset** : draws a box around the related elements
- **legend** : defines a caption for the <fieldset> element

<label> attribute

- ## Example

```
<label for="lname">Last Name</label>
<input type="text" name="lastname" id="lname" value="Wong"
/><br />
First Name
<input type="text" name="fname" id="fname" value="Peter"
/><br />
<label>City <input type="text" name="city" /></label>
```
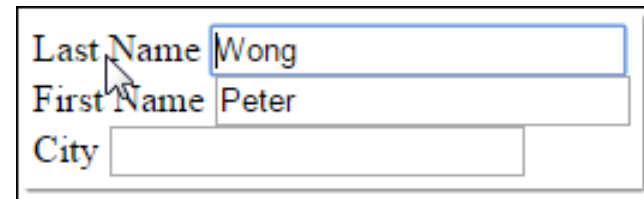
- **for** : specifies which form element a label is bound to
- The <label> element can also wrap the <input> element without the **for** attribute
- The <label> element defines a label for an <input> element and improves *web accessibility*
- Users can *click on the label* to select the associated <input> element
- Some screen reader can read the text of the <label> element to help the visually impaired users (further reading : http://webaim.org/intro/)

Set default value

- ## Example

```
Gender:
<input type="radio" name="gender" value="M"
checked="checked" />Male
<input type="radio" name="gender" value="F" />Female

<select name="education">
  <option>Degree</option>
  <option selected="selected">Higher Diploma</option>
  <option>Form 6</option>
</select>

<textarea rows="4" cols="40">In less 150 words...</textarea>

<input name="username" type="text" value="Chan Tai Man" />
```
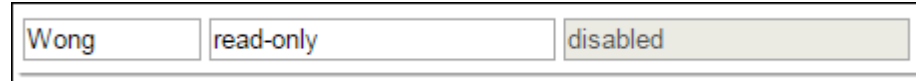
Common attributes for <input> field

- Example

Wong	read-only	disabled

```
<input name="lastname" type="text" size="8" maxlength="15"
value="Wong" />

<input name="firstname" type="text" readonly="readonly"
value="read-only" />

<input name="department" type="text" disabled="disabled"
value="disabled" />
```

- **name** : specifies the name of an input field. Javascript or server-side script uses this name to reference the input field.
- **value** : specifies the initial value for an input field
- **readonly** : specifies that the input field is read only
- **disabled** : specifies the the input field un-usable and un-clickable
- **size** : specifies the size (in number of characters) for the input field
- **maxlength** : specifies the maximum allowed length for the input field
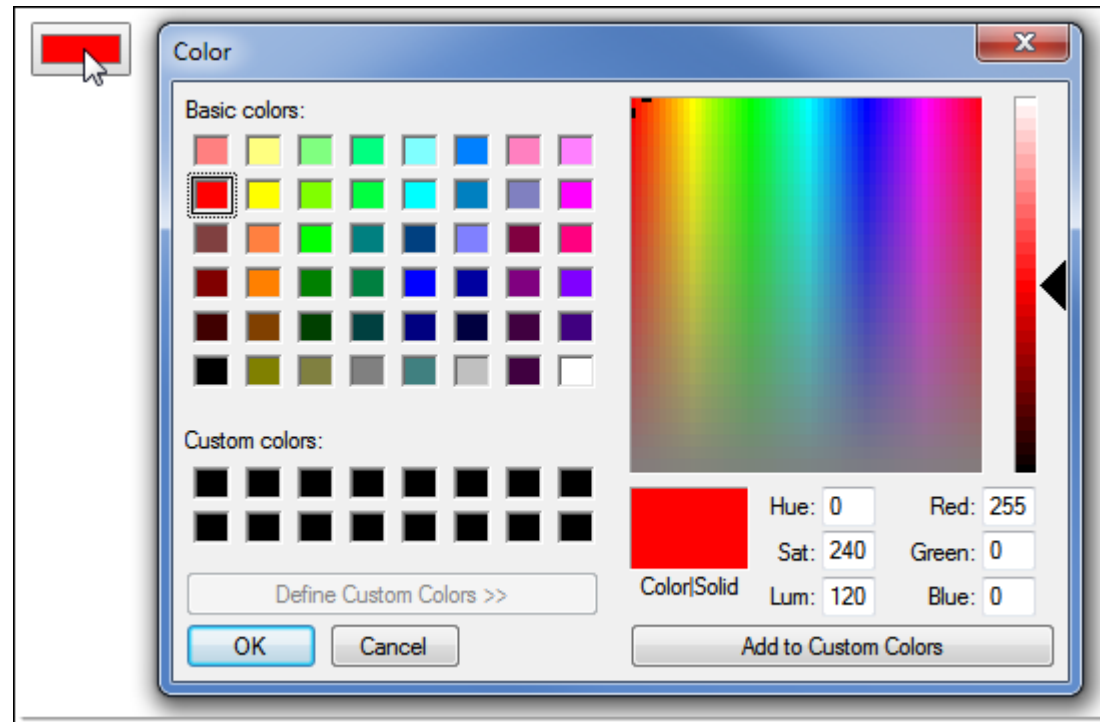
HTML5 : New Input Types

- color

- date
- time
- datetime
- datetime-local
- month
- week

- number
- range

- tel
- email
- url

- search

input type for color data

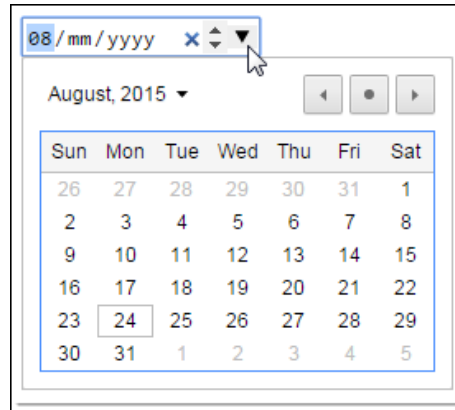- ## Example

```
<input type="color" name="color" />
```

- The input value must be a hexadecimal (hex) value: 3 double digit numbers, starting with a # sign (like #FF8040)

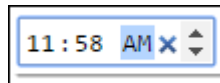- Depending on browser support, a color picker can show up in the input field

input types for date and time data

- ## Example

`<input type="date" />`
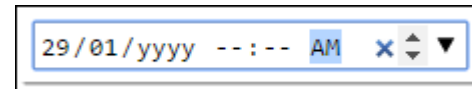
`<input type="time" />`

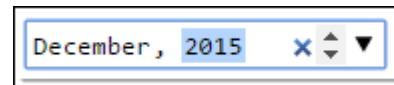`<input type="datetime" />`

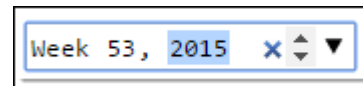`<input type="datetime-local" />`

`<input type="month" />`

`<input type="week" />`

- Depending on browser support, a date/time picker can show up in the input field

input types for numeric value

- ## Example

```
<input type="number" />
```

- You can set restrictions on the numbers

```
<input type="range" min="0" max="10" />
```

- For input fields that should contain a value within a range
- Depending on browser support, the input field can be displayed as a slider control

- ## Online demos :

  - http://www.html5tutorial.info/html5-number.php
  - http://www.html5tutorial.info/html5-range.php

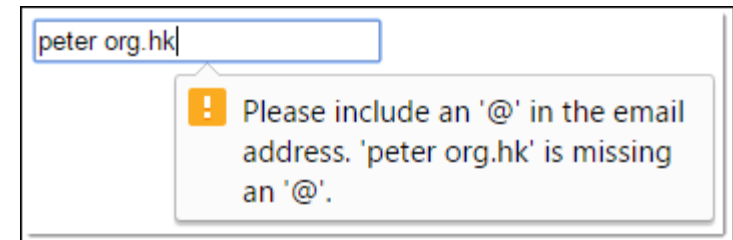# input types for data with format

- ## Example

`<input type="`**`tel`**`" />`

- For input fields that should contain a telephone number
- Due to inherent variances in phone number formats, browsers treat it as a regular, single-line text input field. Can use **`pattern`** attribute to perform data validation.
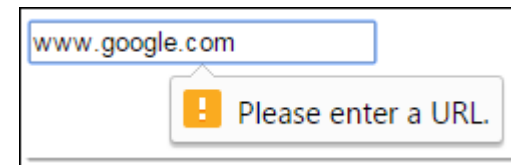- Its purpose is to optimize the keyboard on mobile devices

`<input type="`**`email`**`" />`

- For input fields that should contain an e-mail address

> peter org.hk|
>
> ! Please include an '@' in the email address. 'peter org.hk' is missing an '@'.

`<input type="`**`url`**`" />`

- For input fields that should contain a URL address

> www.google.com
>
> ! Please enter a URL.
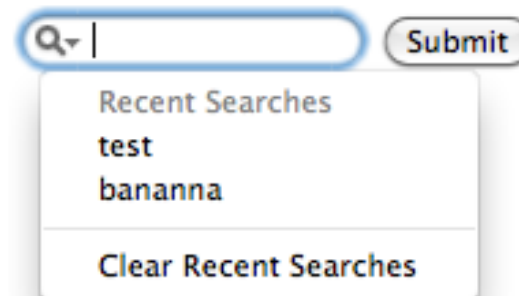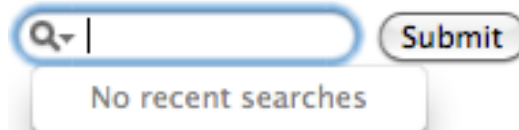
input type for searching data

- ## Example

```
<input type="search" results="2" />
```

- For search fields (a search field behaves like a regular text field)
- Click on the "cross" to clear the input string (*Chrome*)
- **results** :   specify how many recent searches will appear in this dropdown (*Safari*)

<datalist> element : shows predefined options

- ## Example

```
<input list="campus" />
<datalist id="campus">
  <option value="Tsing Yi" />
  <option value="Chain Wan" />
  <option value="Tuen Mun" />
</datalist>
```
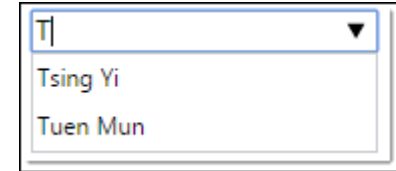
- **datalist** : specifies a list of pre-defined options and provide an autocomplete feature for an <input> element
- **list** : this attribute binds to the <datalist> element

<output> element : represents the result of a calculation

- Example

```
<form
oninput="total.value=parseInt(a.value)+parseInt(b.value)">
  0<input type="range" id="a" />100
  +<input type="number" id="b" value="50" />
  =<output name="total" for="a b" />
</form>
```

- **output** : represents the result of a calculation
- **for** : A space-separated list containing the IDs of the elements whose values went into the calculation.
- **oninput** : execute a JavaScript when a user changes the input value

Online Learning Resources

- `https://www.w3schools.com/html/`
- `https://www.tutorialspoint.com/html/`
- `http://htmldog.com/guides/html/beginner/`

- `Online Editor:`
  - `http://cssdeck.com/labs`