

Lecture 04

# Basic Web Development

HTML & CSS

by K Mohamed Aasik

# About Our Company

## Clevonex



- We elevate brands through world class digital experiences.
- We design, build and maintain software solution for our customers.
- Let's confidently scale your business process worldwide.
- We elevate brands through world class digital experiences.



info@clevonex.tech



+94 705500333



<https://clevonex.tech/>



<https://www.linkedin.com/company/clevonex/>

# About Me



## K Mohamed Aasik

full-Stack Engineer, BIT(R), NDICT(R), Dip  
in HRM, ICTT, Dip in Psychology



aasikmohamed782@gmail.com



+94 75 944 0067



<https://www.sayatmeaasik17.netlify.app>



[linkedin.com/in/mohamedaaasik](https://www.linkedin.com/in/mohamedaaasik)



# Last lecture recap

- HTML Tables
- HTML Lists
- HTML pre tag
- Block and Inline Elements
- HTML Div Element
- HTML Class Attributes
- HTML Head Elements

*Let's clarify all doubts at last of this lecture!*



# HTML Forms

```
<form>
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" value="Vinojan"><br><br>
  <label for="sex">Sex:</label>
  <input type="radio" name="sex" id="male" value="male">
  <label for="male">Male</label>
  <input type="radio" name="sex" id="female" value="female">
  <label for="female">Female</label> <br><br>
  <label for="country">Country: </label>
  <select name="country" id="country">
    <option>Select an option</option>
    <option value="nepal">Nepal</option>
    <option value="usa">USA</option>
    <option value="australia">Australia</option>
  </select><br><br>
  <label for="message">Message:</label><br>
  <textarea name="message" id="message" cols="30" rows="4"></textarea><br><br>
  <input type="checkbox" name="newsletter" id="newsletter">
  <label for="newsletter">Subscribe?</label><br><br>
  <input type="submit" value="Submit">
</form>
```

Name:

Sex: ☐ Male ☐ Female

Country:

Message:

☐ Subscribe?



# HTML Form Elements

`<input>`

`<label>`

`<select>`

`<textarea>`

`<button>`

`<fieldset>`

`<legend>`

`<datalist>`

`<output>`

`<option>`

`<optgroup>`





# Input Types

`<input type="button">`

1. `text` - creates a single-line text fields (default)
2. `button` - creates a button with no default functionality
3. `checkbox` - creates a checkbox
4. `color` - creates a color picker
5. `date` - creates a date picker
6. `datetime-local` - creates a date and time picker
7. `email` - creates an input field that allows the user to input a valid email address
8. `file` - creates an input field that lets the user upload a file or multiple files
9. `hidden` - creates an invisible input field
10. `image` - creates a button using an image



# Input Types

`<input type="password">`

11. `month` - creates an input field that lets the user enter month and year
12. `password` - creates an input field that lets the user enter information securely
13. `radio` - creates a radio button
14. `range` - creates a range picker from which the user can select the value
15. `reset` - creates the button which clears all the form values to their default value
16. `search` - allows user to enter their search queries in the text fields
17. `submit` - allows user to submit form to the server
18. `tel` - defines the field to enter a telephone number
19. `time` - creates an input field that accepts time value
20. `url` - lets the user enter and edit a URL
21. `week` - lets the user pick a week and a year from a calendar





# Input Attributes

- id & name
- value
- readonly
- disabled
- size
- maxlength
- min & max
- placeholder
- required
- step
- autofocus
- height & width
- autocomplete
- multiple



# HTML Form Actions

## GET vs POST methods

GET	POST
Data sent with the GET method is visible in the URL.	Data sent with the POST method is not visible.
GET requests can be bookmarked.	POST requests can't be bookmarked.
GET requests can be cached.	POST requests can't be cached.
GET requests have a character limit of <b>2048</b> characters.	POST requests do not have a limit.
Only ASCII characters are allowed in GET requests.	All data is allowed in POST request



See you in the  
**Next class !**