

6. Bootstrap 5

- 1) What and Why is Bootstrap?
- 2) Containers
- 3) Grid
- 4) Typography & Color
- 5) Margin & Padding
- 6) Buttons & Button Group
- 7) Badges & Icons
- 8) Listing
- 9) Cards
- 10) Dropdowns
- 11) Collapse
- 12) Navigation
- 13) Form
- 14) Carousel
- 15) Modal
- 16) Tooltip & Popover & Toast

1) What and Why is bootstrap?

Bootstrap ဆိုတာကတော့ Css နဲ့ JavaScript Framework တခုဖြစ်ပါတယ်။
Responsive websiteတွေအတွက်အသုံးပြုကြပါတယ်။ကြိုတင်ရေးသားထားတဲ့
navbar,modal,button.., စတဲ့ pre-made components တွေကိုအသုံးပြုနိုင်ပါတယ်။
ဒါအပြင် Layout တွေခွဲတဲ့အခါမှာလည်း Responsive ဖြစ်ဖို့အတွက်
columns တို့ grid စတဲ့ layout system ပါဝင်ပါတယ်။

Bootstrap ကိုသုံးရတဲ့ရည်ရွယ်ချက်တွေကတော့ အများကြီးရှိပါတယ်။

1. အချိန်ကုန်သက်သာစေခြင်း၊
2. pre-made component တွေဖြစ်လို့ သုံးရလွယ်ကူခြင်း၊
3. Responsive ဖြစ်တဲ့ Grid System ရှိခြင်း၊
4. မညှိသည့် Modern Browser တွေမှာမဆိုအသုံးပြုနိုင်ခြင်း၊
5. Model, Carousel စတဲ့ plugin တွေပါဝင်တဲ့အတွက် ဖန်တီးဖို့လွယ်ကူခြင်း၊
6. Customize လုပ်ရလွယ်ကူခြင်း၊
7. Open Source စတဲ့အချက်တွေကြောင့် Css framework ဖြစ်တဲ့ Bootstrap ကိုသုံးရတာဖြစ်ပါတယ်။

2) Containers

Containers ကတော့ သူထဲမှာပါဝင်နေတဲ့ elements တွေကို မတူညီတဲ့ device တွေရဲ့ viewport တွေအလိုက် screen view အတွင်းမှာပေါ်နိုင်အောင် auto လိုက်ညှိပေးတဲ့ bootstrap ရဲ့ basic layout တခုဖြစ်ပါတယ်။ containers တွေကို nested သုံးနိုင်ပေမယ့် layout အများစုကတော့ အသုံးပြုဖို့မလိုအပ်ပါဘူး။

Containers ရဲ့ Viewport တွေဖြတ်ပေးတဲ့ breakpoint တွေကတော့

Breakpoint	Class infix	Dimensions
X-Small	<i>None</i>	<576px
Small	<i>sm</i>	≥576px
Medium	<i>md</i>	≥768px
Large	<i>lg</i>	≥992px
Extra large	<i>xl</i>	≥1200px
Extra extra large	<i>xxl</i>	≥1400px

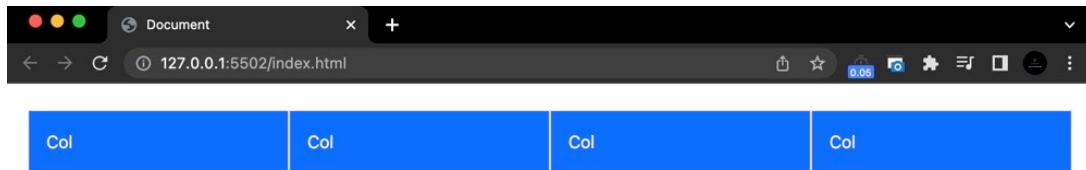
	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
<i>.container</i>	100%	540px	720px	960px	1140px	1320px
<i>.container-sm</i>	100%	540px	720px	960px	1140px	1320px
<i>.container-md</i>	100%	100%	720px	960px	1140px	1320px
<i>.container-lg</i>	100%	100%	100%	960px	1140px	1320px
<i>.container-xl</i>	100%	100%	100%	100%	1140px	1320px
<i>.container-xxl</i>	100%	100%	100%	100%	100%	1320px
<i>.container-fluid</i>	100%	100%	100%	100%	100%	100%

3) Grid

Grid ဆိုတာကတော့ Layout တွေကို ပိုင်းပေးတဲ့ စနစ်တခုဖြစ်ပါတယ်။ Bootstrap ရဲ့ Grid System မှာဆိုရင်တော့ Flexbox ကို သုံးပြီး columnပေါင်း၁၂နဲ့ပိုင်းခြားထားပါတယ်။

Basic Grid

```
<div class="row">
  <div class="col bg-primary text-white p-3 border">Col</div>
  <div class="col bg-primary text-white p-3 border">Col</div>
  <div class="col bg-primary text-white p-3 border">Col</div>
  <div class="col bg-primary text-white p-3 border">Col</div>
</div>
```



အပေါ်ကပုံမှာတော့ col ဆိုတဲ့ class ကိုသုံးထားပါတယ်။ col default ဖြစ်တာကြောင့် Responsive အနေနဲ့အလုပ်လုပ်ပေးမှာဖြစ်ပါတယ်။

Column Width

Column ရဲ့ width ကိုလိုသလောက်ထားချင်တဲ့အခါသုံးပါတယ်။ col-1 ကနေ col-12 ထိသုံးလို့ရပါတယ်။

```
<div class="row">
  <div class="col-6 bg-primary text-white p-3 border">Col 6</div>
  <div class="col-3 bg-primary text-white p-3 border">Col 3</div>
  <div class="col-2 bg-primary text-white p-3 border">Col 2</div>
  <div class="col-1 bg-primary text-white p-3 border">Col 1</div>
</div>
```



ဒီလို col width ကိုသတ်မှတ်ပေးလိုက်ခြင်းဖြင့် ဘယ်viewport မှာမဆို သတ်မှတ်ထားတဲ့ col width အတိုင်းဖြစ်နေမှာ ဖြစ်ပါတယ်။

Responsive Column

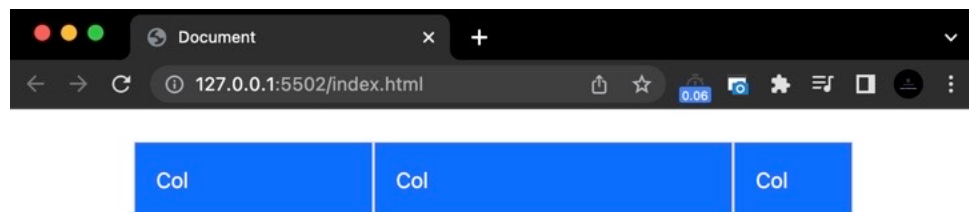
Viewport တွေပြောင်းသွားတဲ့screen size အလိုက် col တွေသတ်မှတ်ချင်တဲ့အခါ Responsive Column တွေကိုသုံးပါတယ်။ဥပမာအနေနဲ့ small screen size တွေမှာဆို col width တွေကတမျိုး၊ large screen တွေမှာဆို col width တမျိုးစီ စတဲ့ screen size တမျိုးစီမှာ မတူညီတဲ့ width တွေကိုပေးပြီးလည်း အသုံးပြုနိုင်ပါတယ်။

```
<div class="row">
  <div class="col-md-8 col-sm-4 col-4 bg-primary text-white p-3
border">Col</div>
  <div class="col-md-2 col-sm-6 col-4 bg-primary text-white p-3
border">Col</div>
  <div class="col-md-2 col-sm-2 col-4 bg-primary text-white p-3
border">Col</div>
</div>
```

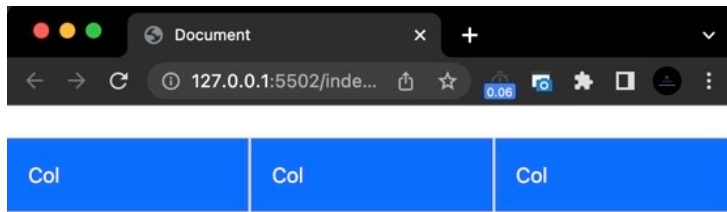
Medium Screen



Small Screen



Under Small Screen



Column offset

Column တွေကို ညာဘက်ကို ရွေ့ချင်တဲ့အခါ သုံးပါတယ်။ offset-*-1 သည်

Column ကို ညာဘက်တကွက်စာရွေ့မယ်ဆိုတဲ့အဓိပ္ပါယ်ဖြစ်ပါတယ်။

* ကတော့ screen break point ဖြစ်ပါတယ်။

```
<div class="row">
```

```
  <div class="col-md-8 offset-md-2 bg-primary text-white p-3 border">Col</div>
```

```
  <div class="col-md-1 bg-primary text-white p-3 border">Col</div>
```

```
  <div class="col-md-1 bg-primary text-white p-3 border">Col</div>
```

```
</div>
```



4) Typography & Color

Typography

■ Heading

.h1 ကနေ .h6 ထိရှိပြီးတော့ စာသားတွေရဲ့အရွယ်အစားကို html tag header တွေရဲ့size အတိုင်းအသုံးပြုနိုင်ပါတယ်။

■ Display

.display-1 ကနေ .display-6 ထိရှိပြီး heading class တွေထက်ကြီးပါတယ်။bold effect လည်း မပါဝင်ပါဘူး။

■ Lead

.lead

Paragraph တွေအတွက်အသုံးပြုပါတယ်။ပုံမှန်စာသားအရွယ်အစားထက် အနည်းငယ်ကြီးပါတယ်။

■ Highlight

<mark>

စာသားတခုကို highlight အဖြစ်ပြချင်တဲ့အခါသုံးပါတယ်။

■ Text alignment

.text-start,.text-center,.text-end

စာသားတွေရဲ့Alignment တွေချိန်ချင်တဲ့အခါသုံးပါတယ်။

■ Text wrapping

.text-wrap

စာသားတွေကို wrap လုပ်ချင်တဲ့အခါသုံးပါတယ်။ nowrap ဆိုရင်တော့

စာသားတွေဟာပေးထားတဲ့width ထက်ဘောင်ကျော်ပြီးပေါ်နေမှာပါ။

■ Word break

.text-break

Space မပါတဲ့ စာသားတွေကိုပေးထားတဲ့widthထက်ကျော်မသွားပဲ

တလိုင်းဆင်းပြီးပေါ်ချင်တဲ့အခါသုံးပါတယ်။

■ Font size

.fs-1 ကနေ .fs-6 ထိရှိပြီးစာလုံးရဲ့အရွယ်အစားကို ချိန်ညှိတဲ့အခါသုံးပါတယ်။

■ Font Style

.fw-bold (bold အတွက်)

.fw-bolder (boldထက်ပိုထူ)

.fw-normal (ပုံမှန် font weight)

.fw-light (font weight နည်း)

.fw-lighter (font weight ပိုနည်း)

.fst-italic (စာသားအစောင်း)

.fst-normal (ပုံမှန်စာသား)

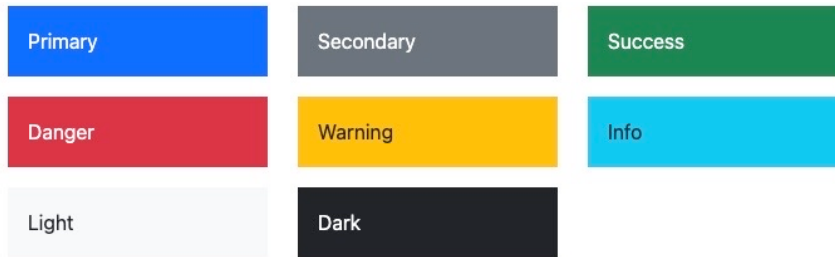
■ Text Decoration

.text-decoration-underline (စာသားကို Underlineထားတဲ့အခါ)

.text-decoration-line-through (စာသားကိုအလယ်ကနေ ဖြတ်တဲ့အခါ)

.text-decoration-none (စာသားကို ဘာeffect မှမထည့်ချင်တဲ့အခါ a tag နဲ့တွဲသုံးပါတယ်)

Color



Bootstrap ရဲ့ pre defined colorတွေဖြစ်ပါတယ်။

စာသားအတွက်ဆိုရင်တော့

.text-primary
.text-secondary
.text-success
.text-danger
.text-warning
.text-info
.text-light
.text-dark
.text-white
.text-muted

Background အတွက်ကတော့

.bg-primary
.bg-secondary
.bg-success
.bg-danger
.bg-warning
.bg-info
.bg-light
.bg-dark
.bg-white
.bg-muted

5) Margin & Padding

Margin,Padding & gap အတွက်လည်း Bootstrap မှာသတ်မှတ်ထားတဲ့ properties တွေရှိပါတယ်။

.m သည် margin အတွက်ဖြစ်ပြီး **.p** သည် padding အတွက်ဖြစ်ပါတယ်။

Side တွေအလိုက် ပေးလို့ရပါတယ်။

- **.t** သည် margin top or padding top
- **.b** သည် margin bottom or padding bottom
- **.s** သည် margin-left or padding-left
- **.e** သည် margin-right or padding-right
- **.x** သည် left နဲ့ right နှစ်ခုစလုံးအတွက်ဖြစ်ပြီး
- **.y** သည် top နဲ့ bottom ဖြစ်ပါတယ်။

ပေးနိုင်တဲ့ size တွေကတော့ 0 ကနေ 5 , auto တို့ပါဝင်ပါတယ်။

Auto ကတော့ margin: auto အနေနဲ့ အလုပ်လုပ်မှာပါ။

```
<span class="bg-primary text-white me-2 p-4">
```

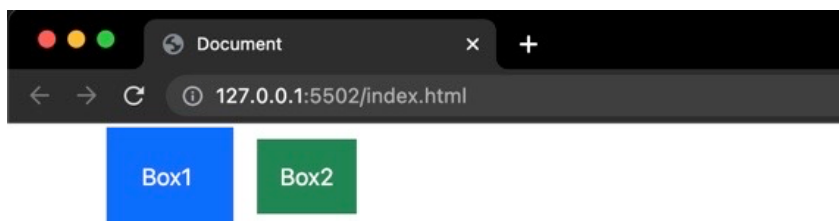
Box1

```
</span>
```

```
<span class="bg-success text-white ms-2 p-3">
```

Box2

```
</span>
```

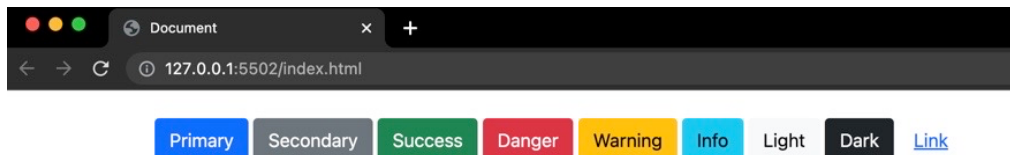


6) Buttons & Button Group

Basic Buttons

Bootstrap ရဲ့ button မှာတော့ pre-defined color တွေအတိုင်း button အရောင်တွေနဲ့ ရှိပါတယ်။

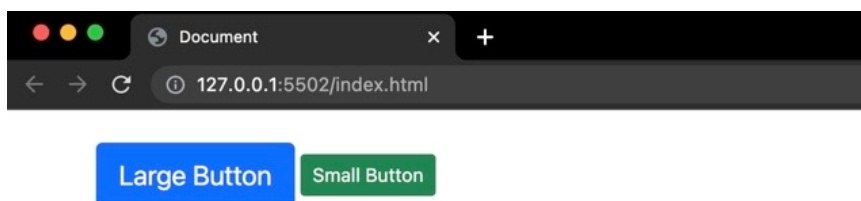
```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-link">Link</button>
```



Button Size

Button Size မှာတော့ .btn-lg နဲ့ .btn-sm ဆိုပြီး အရွယ်အစားနှစ်ခု သတ်မှတ်ထားပါတယ်။

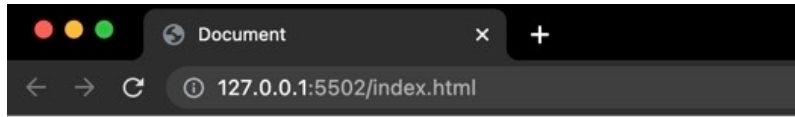
```
<button class="btn btn-lg btn-primary">Large Button</button>
<button class="btn btn-sm btn-success">Small Button</button>
```



Button Group

Button တွေကိုတစုတည်းgroupလုပ်ပြီး ပြစေချင်တဲ့အခါသုံးပါတယ်။

```
<button class="btn btn-primary">Button1</button>  
<button class="btn btn-success">Button2</button>  
<button class="btn btn-warning">Button 3</button>
```



7. AJAX & Additional Lectures

- 1) Why AJAX?
- 2) Request & Response
- 3) AJAX (JavaScript)
- 4) AJAX (jQuery)
- 5) Cookies, LocalStorage, Session
- 6) Additional
- 7) Useful Js libraries

1) Why AJAX?

Client - Server communication တွေမှာ client ကနေ server ဆီကို ဆက်သွယ်ဖို့ (HTTP request/response) ငှမို့ရှိပါတယ်။

- Link
- Form
- Ajax(JavaScript)
- Cookies

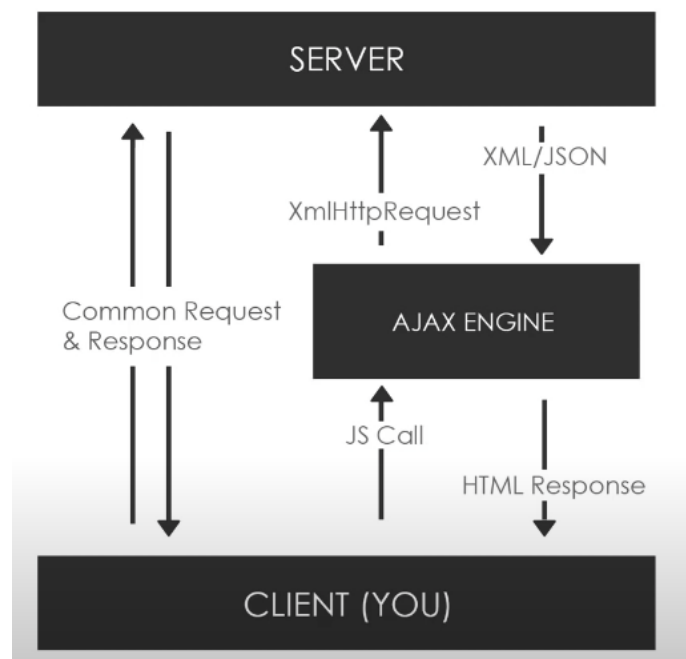
Link နဲ့ Form tag တွေဟာ client ကနေ server ကို Http request လုပ်ပြီး server ဘက်ကနေ response မပြန်ခင်ကြားကာလမှာ page loading ဖြစ်နေပြီး Response ကိုလုပ်ခံရရှိတဲ့အခါ

page ဟာ refresh ဖြစ်သွားပါတယ်။

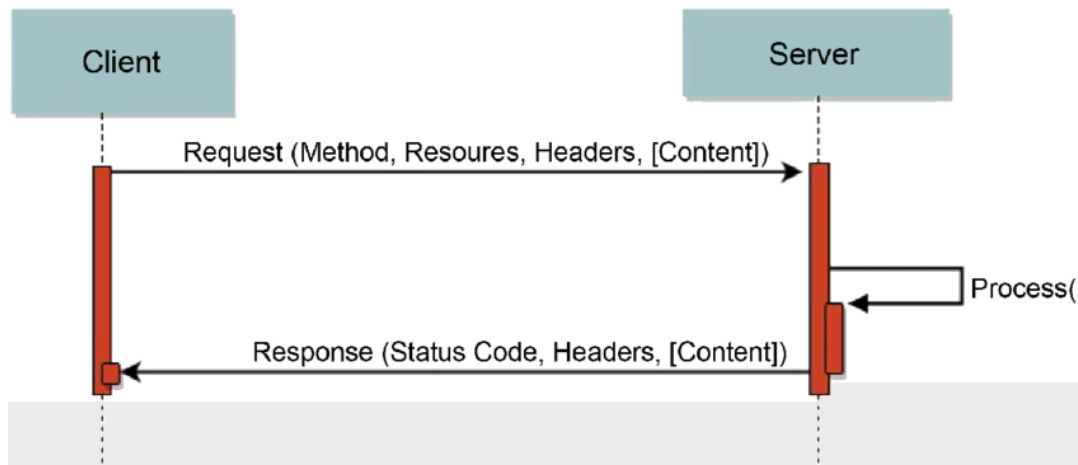
Page transaction မလိုအပ်တဲ့ Web page မှာဆိုရင် Client ကနေ Server ကို Communication လုပ်တိုင်း refresh ဖြစ်နေမှာဖြစ်တာကြောင့် current state ကို ထိန်းပေးထားနိုင်မှာမဟုတ်တော့ပါဘူး။

အဲ့အတွက် Ajax ကိုသုံးပါတယ်။

Ajax က မလိုအပ်တဲ့ page refresh ကို မဖြစ်အောင် server နဲ့ communication လုပ်ပေးနိုင်ပါတယ်။



2) Request & Response



Client ကနေ data တွေကို server ဆီကနေရယူချင်တဲ့အခါ Server ဆီကို request statement တွေပို့ရပါတယ်။ request methods ဇယားရှိတဲ့အထဲ အသုံးများတဲ့ methods နှစ်ခုကတော့ Get နဲ့ Post ဖြစ်ပါတယ်။

Server ဆီက ရတာမှန်သမျှအကုန်ယူမယ်ဆိုရင် Get method ကိုသုံးပါတယ်။

Server ဆီကအကုန်မဟုတ်ပဲ user က input လုပ်လိုက်တဲ့ data တွေကိုကိုင်ညှိတဲ့ result data တွေလိုချင်တဲ့အခါ Post ကိုအသုံးပြုပါတယ်။

Request တခုမှာအနည်းဆုံးပါဝင်ရမှာတွေကတော့

Request

- url
Request လုပ်ချင်တဲ့ server destination
- method
Get သို့မဟုတ် Post
- resource
Get ဆိုရင်တော့ ထည့်စရာမလိုပါဘူး။ post ဆိုရင်တော့ data ကိုထည့်ပေးရမှာဖြစ်ပါတယ်။

- header

Client နဲ့ server communication လုပ်တဲ့အခါ additional information တွေထည့်ပေးဖို့အတွက်သုံးပါတယ်။

Server ဘက်က response ပြန်လာတဲ့အခါ ပါဝင်တာတွေကတော့

Response

- status Codes

200 : OK

201 : Created

202 : Accepted

203 : Non-Authorize information

204 : No Content

400 : Bad Request

403 : Forbidden

404 : Not Found

405 : Method Not Allow

408 : Request Time Out

414 : URL too long

429 : Too many request

500 : Server Error

503 : Service Unavailable

504 : Gateway timeout

- header

Server ဘက်ကပြန်လာတဲ့ header information

- body

Server ဆီက return ပြန်လာတဲ့ data တွေဖြစ်ပါတယ်။json သို့မဟုတ်

Xml format တွေနဲ့ return ပြန်ပါတယ်။

Cross-origin resource sharing (Cors)

Client ကနေ server ကို request တွေလုပ်တဲ့အခါ server ဘက်က Request လုပ်နိုင်တဲ့ Schema, Domain, port ကိုသတ်မှတ်လို့ရပါတယ်။ သတ်မှတ်ထားတဲ့ အတိုင်းမဟုတ်ပဲ တခြား origin ကနေ Access လုပ်တဲ့အခါ Cors ဆိုတဲ့ origin မတူညီမှုတွေဖြစ်လာပါတယ်။အဲလိုserver ဘက်မှာ origin ကန့်သတ်ဖို့အတွက် Access-Control-Allow-origin ကို အသုံးပြုပါတယ်။

Access-Control-Allow-origin

တူညီတဲ့ domain သို့မဟုတ် request လုပ်နိုင်တဲ့ domain တွေကို သတ်မှတ်ထားနိုင်ပါတယ်။အားလုံးကိုဝင်စေချင်တဲ့အခါ * သတ်မှတ်ပေးနိုင်ပါတယ်။

3) AJAX (JavaScript)

▪ XMLHttpRequest()

```
// Create an XMLHttpRequest object
const xhttp = new XMLHttpRequest();

// Define a callback function
xhttp.onload = function() {
  // Here you can use the Data
}

// Send a request
xhttp.open("GET", "ajax_info.txt");
xhttp.send();
```

▪ fetch

```
fetch('http://example.com/songs')
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(err => console.error(err));
```

4) AJAX (jQuery)

Jquery နဲ့ရေးနိုင်တဲ့ Method တွေကတော့

- **load**

Common html ရေးထားပြီး လိုချင်တဲ့ tag မှာ ပြန်ယူတဲ့အခါအသုံးပြုပါတယ်။

Syntax

```
$( 'selector' ).load( 'url' , callback )
```

- **get**

Get method အနေနဲ့သုံးရတဲ့ http request တွေမှာသုံးပါတယ်။

Syntax

```
$.get( 'url' , callback )
```

- **post**

Post method အနေနဲ့ သုံးတဲ့ http request တွေမှာ သုံးပါတယ်။

Syntax

```
$.post( 'url' , callback )
```

- **getJSON**

Json return ပြန်တဲ့ response တွေအတွက် get Json ကိုသုံးနိုင်ပါတယ်။

Request method ကတော့ get ဖြစ်ပါတယ်။

Syntax

```
$.getJSON( 'url' , callback )
```

- **ajax**

အပေါ်ကလေ့လာခဲ့တဲ့ Method တွေထက်ပိုပြီး customization

ဖြစ်အောင်ရေးနိုင်ဖို့ ၊ ajax ကိုသုံးနိုင်ပါတယ်။ Ajax မှာဆိုရင်တော့

Settingတွေများစွာရှိပါတယ်။အဲ့ဒီထဲက အသုံးဝင်တဲ့ setting တွေကတော့

Syntax

```
$.ajax('url' ,[setting])
```

-setting

Cache (true,false)

Cache သိမ်းထားမလား၊မထားဘူးလားဆိုဖြတ်တဲ့အခါသုံးပါတယ်။

Head နဲ့ Get request တွေမှာသာလျှင်အလုပ်လုပ်နိုင်ပါတယ်။

ContentType

ContentType တွေသတ်မှတ်တဲ့အခါသုံးပါတယ်။

ContentType များစွာရှိပါတယ်။

Application

application/EDI-X12

application/EDIFACT

application/javascript

application/octet-stream

application/ogg

application/pdf

application/xhtml+xml

application/x-shockwave-flash

application/json

application/ld+json

application/xml

application/zip

application/x-www-form-urlencoded

Audio

- audio/mpeg
- audio/x-ms-wma
- audio/vnd.rn-realaudio
- audio/x-wav

Image

- image/gif
- image/jpeg
- image/png
- image/tiff
- image/vnd.microsoft.icon
- image/x-icon
- image/vnd.djvu
- image/svg+xml
- Multipart multipart/mixed
- multipart/alternative
- multipart/related (using by MHTML (HTML mail).)
- multipart/form-data

Text

- text/css
- text/csv
- text/html
- text/javascript (obsolete)
- text/plain
- text/xml

Video

- video/mpeg
- video/mp4
- video/quicktime
- video/x-ms-wmv
- video/x-msvideo
- video/x-flv
- video/webm

VND

application/vnd.oasis.opendocument.text
application/vnd.oasis.opendocument.spreadsheet
application/vnd.oasis.opendocument.presentation
application/vnd.oasis.opendocument.graphics
application/vnd.ms-excel
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
application/vnd.ms-powerpoint
application/vnd.openxmlformats-officedocument.presentationml.presentation
application/msword
application/vnd.openxmlformats-officedocument.wordprocessingml.document
application/vnd.mozilla.xul+xml

Data

Server ကိုပို့ချင်တဲ့ Data ဖြစ်ပါတယ်။

Datatype

Data ရဲ့ format ဥပမာ json, xml,html စသည်ဖြင့်ရှိပါတယ်။

Headers

Server ဆီdata ပို့တဲ့အခါ data အပြင် တခြား additional info တွေပါ ထည့်ချင်တဲ့အခါ header tag မှာ ထည့်ပေးနိုင်ပါတယ်။

Method

Server ကို request လုပ်တဲ့ method ဖြစ်ပါတယ်။
Get,Post,Put...,စသည်ဖြင့်ရှိပါတယ်။

Error

Request တွေ fail ဖြစ်တဲ့အခါ catch ဖမ်းပေးတဲ့ method တခုဖြစ်ပါတယ်။

Success

Request success ဖြစ်တဲ့အခါ catch ဖမ်းပေးတဲ့ method ဖြစ်ပါတယ်။

Status Code

Server ဆီက return ပြန်လာတဲ့ status code ကို ကြည့်ပြီး လိုချင်တဲ့ processတွေကိုသတ်မှတ်တဲ့အခါသုံးပါတယ်။

Timeout

Server ဆီ request

လုပ်တဲ့အချိန်ကိုသတ်မှတ်တာဖြစ်ပါတယ်။

ဘာမှမသတ်မှတ်ထားရင်တော့ infinity ဖြစ်ပါတယ်။

Url

ကိုယ်ပို့ချင်တဲ့ server ပတ်လမ်းကြောင်းဖြစ်ပါတယ်။

5) Cookies, LocalStorage, Session

User အသုံးပြုတဲ့ browser တွေမှာ data သိမ်းထားနိုင်တဲ့ storage အနေနဲ့ သုံးခုရှိပါတယ်။ သိမ်းထားတဲ့ storage ကို Devtools ရဲ့ Application tag မှာ ကြည့်နိုင်ပါတယ်။

Cookies vs. Local Storage vs. Session Storage			
	Cookies	Local Storage	Session Storage
Capacity	4kb	10mb	5mb
Browsers	HTML4 / HTML 5	HTML 5	HTML 5
Accessible from	Any window	Any window	Same tab
Expires	Manually set	Never	On tab close
Storage Location	Browser and server	Browser only	Browser only
Sent with requests	Yes	No	No

Cookies

Data storage 4 KB ရပြီး Expired ကို လိုသလိုသတ်မှတ်ထားနိုင်ပါတယ်။

Set Method

```
document.cookie = "key=value; expires="+new  
Date(year,month,day).toUTCString()
```

Local Storage

Data storage 10 MB ထိရပြီးတော့ မဖျက်သ၍
သိမ်းထားမှာဖြစ်ပါတယ်။

Set Method

```
localStorage.setItem("key","value")
```

Get Method

```
localStorage.getItem("key")
```

Remove Method

```
localStorage.removeItem("key")
```

Session Storage

Data storage 5 MB ထိရပြီး တူညီတဲ့ tag မှာပဲ Access

လုပ်ခွင့်ရမှာဖြစ်ပါတယ်။ tag ကိုပိတ်လိုက်တဲ့အခါသိမ်းထားတဲ့ data တွေက ပျက်သွားမှာဖြစ်ပါတယ်။

Set Method

```
sessionStorage.setItem("key","value")
```

Get Method

```
sessionStorage.getItem("key")
```

Remove Method

```
sessionStorage.removeItem("key")
```