natUIve framework

Element	Code	Options

Grid	`<div class=row>`
	
	   `<div> … </div>`
	   …
	
	`</div>`
	
	Options
	
	• Columns: just a `div`, .col3, .col23, .col34 etc.
	• `.contain` constrains to standard page width
	• Columns have equal width automatically. If a column gets explicit width, the others divide the rest equally among themselves. When columns become too narrow for one row, they are wrapped to another row automatically.
	• `.row.half` for 2-column row on mobile
	• `.row.border` for column borders
	• use `.pad` or `.pad-x2` for padding; `.unpad` for enlarging with negative margin
	• embed `.box` in cells for boxes with different background, border etc.
	• `.row.border-all` for borders around each cell
		• cell alignment classes: `.align-right`, `.align-center`, `.align-top`, `.align-middle`, `.align-bottom`. Vertically-Aligned elements must be wrapped inside an extra div. `.row` inherits alignment
		• `.align-children` inherits alignment
		• `.row > .vertical-row` – vertical row as tall as the parent row with equal-height full-width children
		• `.row.no-spacing` glues the cells together (CSS variables support required)
		• `.row.no-padding` columns without padding (CSS variables support required)
		• `.row.no-margin` for no gutter between columns (CSS variables support required)
		• `.row > .full-mobile-width` for full width columns on mobile

Mobile/Desktop Content	`<div class=narrow-screen>`
	   …
	`</div>`
	
	`<div class=wide-screen>`
	   …
	`</div>`	

natUIve framework

Element	Code	Options

Structure	`<div class=header-and-banner>`
	
	    `<header>`
	      …
	    `</header>`
	
	    `<div class=banner>`
	
	    `</div>`
	
	`</div>`
	
	`<main id=content>`
	
	    `<section class=contain>`
	      …
	    `</section>`
	    …
	`</main>`
	
	`<footer>`
	
	  …
	
	`</footer>`
		• `.header-and-banner.viewport` sets header/banner to viewport height. `.viewport.push-up` to move the content upwards for a better visual balance
		• `section` clears inner margins at top/bottom
		• `header.fixed` for fixed header on top on narrow screens

natUIve framework

Element	Code	Options

Modal Window	`<a href=page.html#container class=modal>`
	
	`    Open in modal window`
	
	`</a>`
		• `#container` is recommended to avoid HTML head parts
		• `openFullWindow(HTML)` shows the HTML code in a modal window.
		• `openFullWindow(HTML, '0% { … } 100% { … }')` for custom animation, automatically reversed on close
		• `a.modal[data-anim]` parameter specifies custom animation
		• `a.limited` for limited width/height modal window

natUIve framework

Element	Code	Options

Slider	`<div class=slider data-autoslide=5 data-duration=.5>`
	
	    `<div> … </div>`
	     `…`
	
	`</div>`
	
	Options
	
	• `slider.tabs` makes thumbnail navigation instead of numbered one. Tab titles from a `.tab-title` child or a `data-tab_title` attribute. Supports `.top.tabs,` optional `.slider.wrap` for auto-width tab links
	• `data-autoslide=5` slides to next slide after a 5" delay
	• Create sliders dynamically with `makeSlider(el)`
	• `.slider.vertical-thumbnails` option
	• `.slider.vertical` option
	• `.slider.full-window` option
	• `.slider.full-window.headless` option
	• `data-duration=x` option for x seconds of sliding animation
		• `.slider.center` centers slides content
		• `.slider.auto-height` matches slider height to the current slide
		• `.slider.fade` to fade in/out instead of sliding
		• `.vertical.right, .vertical.tabs.right`
		• `.slider.pad` for outside controls
		• `.slider.align-middle` to vertically center slides (requires extra slide content wrapper)
		• To detach nav from slider, prepare a `.slider#id` and `.slider-nav[data-for=id]`
		• url#hash auto slides a slider with `slide#hash` to that one.

natUIve framework

Element	Code	Options

Tooltip	`<span class=tool>`
	
	    `Hover for a tip`
	    `<span class=tip> Info </span>`
	    `…`
	
	`</span>`	

Accordion (Fold)	`<div class=fold>`
	
	    `<div class=label> Label </div>`
	
	    `<input type=checkbox class=trigger>`
	
	    `<div class=content> … </div>`
	
	`</div>`
		• Embedded accordions supported
		• Optionally open by default with `.fold.open`
		• Content's CSS variable `--start-height` truncates it to the specified height.
		• `.fold.horizontal`, needs extra wrapper inside `.content`

Buttons	`<a href=# class=button> Call to Action </a>`	• `.big` option • wrap a group of buttons in `.group`

natUIve framework

Element	Code	Options

Forms	```<form>```
	
	 `<label>`
	 `<span> Description </span>`
	 `<input> // <textarea></textarea>`
	 `</label>`
	
	 `<label>`
	 `<span>Select a file</span>`
	 `<span class=file-input>`
	 `<input type=file>`
	 `<span></span>`
	 `</span>`
	 `</label>`
	
	 `<div class=checkbox>`
	 `<span> Description for all checkboxes</span>`
	 `<label>`
	 `<input type=checkbox> <b class=checkbox-icon> </b>Specific description`
	 `</label>`
	 `…`
	 `</div>`
	
	 `<label>`
	
	 `<span> Select Description </span>`
	 `<select> … </select>`
	
	 `</label>`
	
	 `…`
	 `<input type=submit>`
	
	 `</form>`
		• Description with Tooltip: `<span class=tool>Description <span class=tip> </span>`
		• `<label class=mandatory>`
		• `form.mast` for a balanced layout with labels in left column and inputs on right column
		• `<input type=number data-digits=4>`
		• `form.wide` for 100% wide inputs
		• `form.sticky-submit` positions the submit button to the right/center of the form
		• `form.dynamic` submits without reloading the page and displays the result in a modal window
		• `form { --placeholder_color: #aaa; }` is the custom placeholder colour variable
		• `fieldset[disabled]` is toggled by its previous sibling `.checkbox.condition`
		• Set the placeholder colour with CSS variable — `placeholder_color`
		• `<textarea data-auto …` to automatically adjust the height

natUIve framework

Element	Code	Options

Lightbox	```<div class=lightbox>```
	
	    ``` Click for Image 1 ```
	    ``` <img src=thumbnail.png alt=Thumbnail> ```
	    …
	
	```</div>```
		When clicked, a `.lightbox` child link to image forms a common slider gallery with its siblings.
		• `lightbox.full-screen` option
		• Use `http://url#lightbox_id` to open it automatically on page load
		• `url?slide=n#lightbox_id` opens nth slide of that lightbox
		• `url?image=image_name.jpg#lightbox_id` opens that image's slide
		• `a.video`
		• `a.iframe`
		• `.thumbnails` for thumbnail controls.
		• `.inline` for an inline lightbox.
		• `.inline.thumbnails.outside` for outside-positioned controls.
		• Lightbox with `id` provides links to individual slides
		• `.lightbox[data-anim]` parameter specifies custom animation

Quote	```<p class=quote>```
	    …
	```</p>```	

natUIve framework

Element	Code	Options

Aspect ratio image container	<pre><picture class=aspect style="--
	ratio: Ratio">
	
	    <img src=image.jpg alt=''>
	
	</picture></pre>
		Ratio = x/y. Works only when container is 100% wide.

Masonry CSS-only	<pre><div class=masonry>
	    <div> <p>Column</p> </div>
	     …
	</div></pre>
		CSS columns with vertically optimised cell positioning. Masonry effect (vertical track only).

Tables		`<td data-sort>` makes a column sortable. The `data-sort` value can be unspecified, **asc** or **desc**.

Top navigation	<pre><header>
	    <div class='fold mobile'>
	        <div class=label> <span
	class="burger"> </div>
	        <input type=checkbox
	class=trigger>
	        <div class=content>
	            <nav>
	            …
	            </nav>
	        </div>
	    </div>
	</header></pre>
		Hidden and burgerised on mobile
		• `header.fixed` option

natUIve framework

Element	Code	Options

Drop-down nav	```<nav class=drop>```
	``` <form id=level1 class="nav-form"> </form>```
	``` <form id=level2 class="nav-form"> </form>```
	``` <input type=checkbox class='trigger burger'>```
	``` ```
	``` ```
	``` <input type=radio class=trigger form=level1 name=level1>```
	``` <input type=reset form=level1>```
	``` <a>One```
	``` <b class=sub>```
	``` ```
	``` ```
	``` <input type=radio class=trigger form=level2 name=level2>```
	``` <input type=reset form=level2>```
	``` <a>Middle One```
	``` <b class=sub>```
	``` ```
	``` ```
	``` Bottom One```
	``` ```
	``` ```
	``` ```
	``` ```
	``` ```
	``` ```
	```</nav>```	

natUIve framework

Element	Code	Options

Scrollable nav	`<nav class=scroll>`
	
	`    <ul> … </ul>`
	
	`</nav>`
		• overflowing navigation to scroll horizontally instead of wrap on new lines.

Drop caps	`<p class=drop-cap>`
	`    …`
	`</p>`
		• larger first letter

CSS-only parallax scrolling	`<html class=parallax>`
	`    <head> … </head>`
	`    <body>`
	
	`        <div class=parallax-back> … </div>`
	
	`        <div class=parallax-front> … </div>`
	
	`    </body>`
	`</html>`
		• `.parallax-back` is the back layer, scrolling slower than the front layer `.parallax-front`

Right to left language support	`<body dir=rtl>`	

Cards	`<div class=card>`
	
	`    <div class=head> … </div>`
	`    <div class=content> … </div>`
	
	`</div>`
		• `.head.image` with inline CSS for `background-image`

natUIve framework

Element	Code	Options
Notification bar	```<body> <div class=notify> … </div> …</body>```	• option `.fixed` • create dynamically by `notify('Message')` with optional parameter `'fixed'` • if already present in HTML statically, close action will work • `notify('Text', 'timeout')` to hide it after 2 seconds
Lists	``` … … ```	• `ul.inline` orders items on one line • `ul.indent`
Typography		• option `.big-text`
Copy to clipboard by JavaScript	`copyButton (el, target)`	• `el` is the button element, `target` is the source of the clipboard text
Language selector	```<form class=language><label class=select><select name=l><option value=en>🇬🇧 English</option>…</select></label></form>```	• Navigates to `url?l=en`

natUIve framework

Element	Code	Options
Links	`<a href=# class="span-relative-parent"></a>`	• Overlays a link over its whole parent container

Fixed background section	`<div class=fixed-bg style="background-image: url(image.jpg)">`
	`    <div class=fixed-bg-image></div>`
	`    …`
	`</div>`
		• `.fixed-bg` – element as a window to an image spanning the viewport. Needs inline `background-image`.

Scroll threshold	`<div data-threshold>`
	
	`    …`
	
	`</div>`
		• Adds CSS variable `--threshold` showing how much of the element height has been scrolled up outside of viewport, 0.0 ~ 1.0.
		• `--height` is the element height, for calculations
		• When all of it is above the viewport, the element gets a `.threshold` class