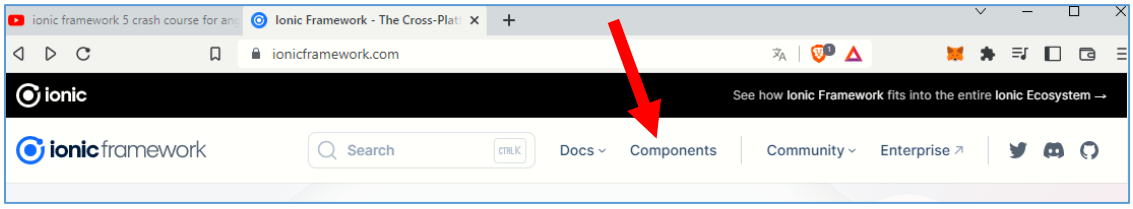
Ionic - Buttons allgemein

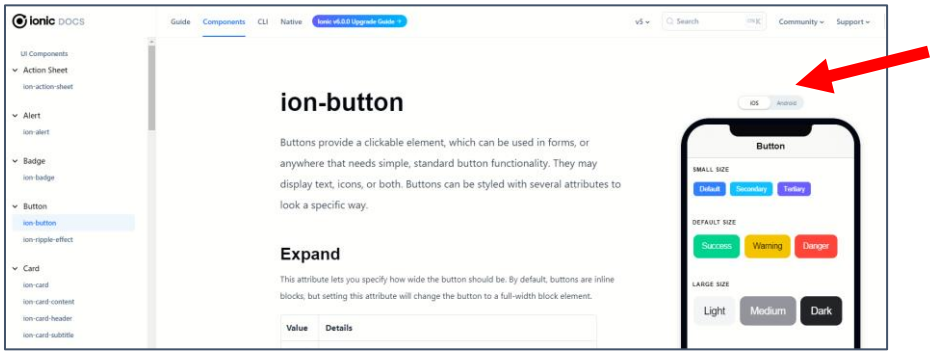
1. Neue Seite „infobutton“ einfügen und im Side-Menü anlegen
2. Aussehen des Buttons ändern inkl. Icons
3. Weiterleitung zu einer anderen Seite
   3a)routerLink=“page-name“
   3b)mit click-Funktion
4. Alert per click() erstellen

Button einfügen

Zum Erstellen von Button betrachte kurz die Doku der Website.

Klicke auf „Components“.

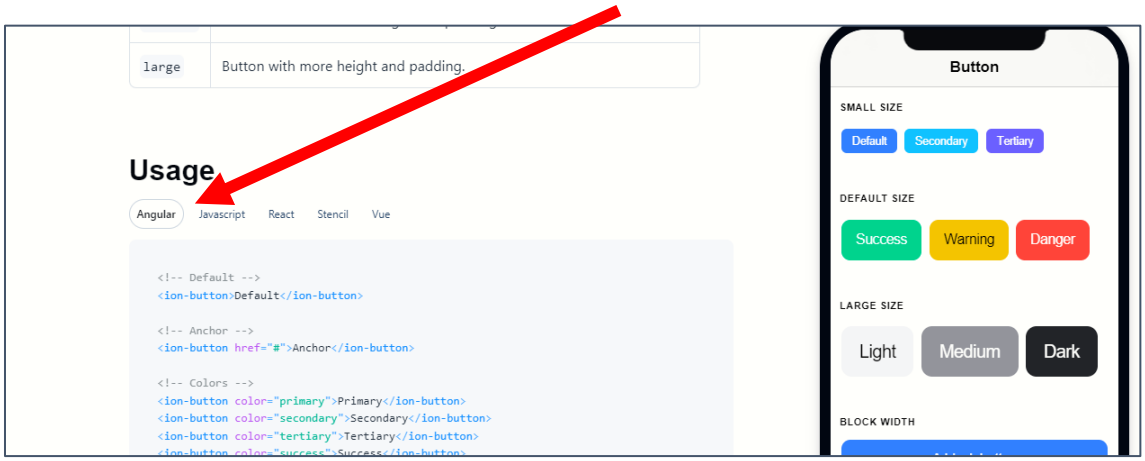




Hier kann man zwischen ios und android Ansicht wählen

Info: https://www.youtube.com/watch?v=qiXcNhXJ_X8 ca. 35 Minuten 20 Sekunden

Beachte, dass man, wenn man Code entnimmt in „Angular“-Modus sich befindet:



Hier findet man viele Eigenschaften für den Button.

1.)Erstelle eine page mit dem Namen „infobutton“

```
ionic g page pages/infobutton
```

Damit man diese dann aufrufen kann, soll im Side-Menü ein Link eingefügt werden:

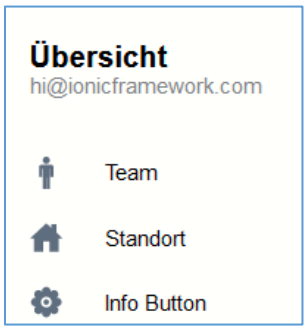
Öffne das Side-Menü mit „app.components.ts“

Füge unterhalb vom Standort eine neue Zeile ein. Wähle ein neues Symbol aus der Website: https://ionic.io/ionicons

```
<ion-icon name="flower"></ion-icon>
```

```
10  public appPages = [
11    { title: 'Team', url: '/team', icon: 'man' },
12    { title: 'Standort', url: '/standort', icon: 'home' },
13    { title: 'Info Button', url: '/infobutton', icon: 'flower' },
```

Ergebnis:



Öffne die „infobutton.page.html“

INFO: Der Button selbst kann drei Füllungen haben:

Value	Details
`clear`	Button with a transparent background that resembles a flat button.
`outline`	Button with a transparent background and a visible border.
`solid`	Button with a filled background. Useful for buttons in a toolbar.

Usage (Anwendung):

```
<!-- Default -->
<ion-button>Default</ion-button>

<!-- Anchor -->
<ion-button href="#">Anchor</ion-button>
```

Der Zusatz „ion“ macht den Button in ionic erst zu einem schönen Button.

Übung:

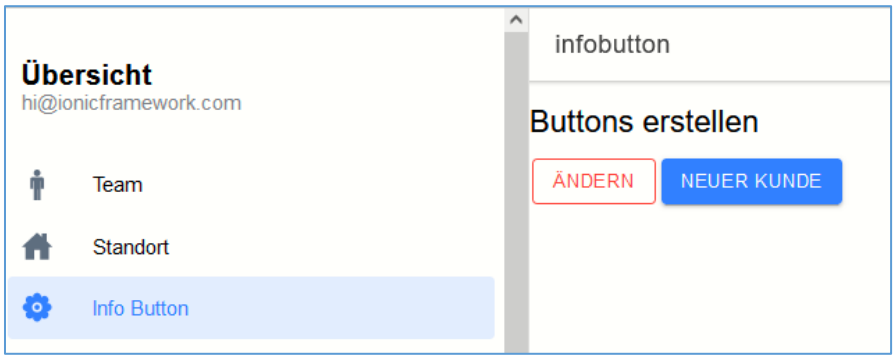
Erstelle im content eine <h1> und darunter zwei Buttons:

<ion-button fill="outline" color="danger">Ändern</ion-button>

<ion-button>Neuer Kunde</ion-button>

```
7   <ion-content>
8     <h2>Buttons erstellen</h2>
9     <ion-button fill="outline" color="danger">Ändern</ion-button>
10    <ion-button>Neuer Kunde</ion-button>
11  </ion-content>
```

Ergebnis:

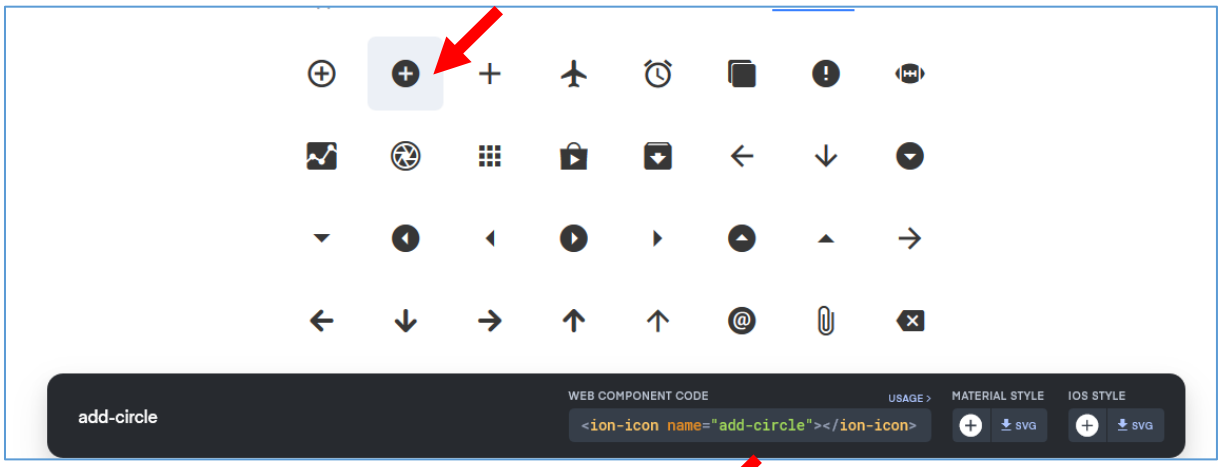


2)Aussehen des Buttons ändern

Im Button „Neuer Kunde“ soll ein Icon eingefügt werden.

Das passende Namenattribut findet man unter https://ionic.io/ionicons

Klicke dort auf ein passendes Icon um unterhalb den Namen zu erfahren:

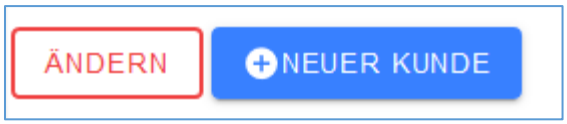



```
9   <ion-button><ion-icon name="add-circle">></ion-icon>Neuer Kunde</ion-button>
```

Man kann auch die Farbe des Icons ändern durch Einfügen der Color:

```
<ion-button><ion-icon slot="start" name="add-circle" color="warning"></io
```

Ergebnis:



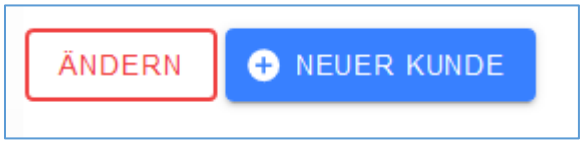
Um einen schöneren Abstand zu erhalten muss man einen „slot“ einfügen.

Daher folgender Code:

```
slot="start"
```

```
<ion-button href=""><ion-icon slot="start" name="add-circle"> </ion-icon>Neuer Kunde</ion-button>
```

Ergebnis:



Der „slot“ dient aber eigentlich für folgendes:

- start: links
- end: rechts

Slots

Name	Description
	Content is placed between the named slots if provided without a slot.
`"end"`	Content is placed to the right of the button text in LTR, and to the left in RTL.
`"icon-only"`	Should be used on an icon in a button that has no text.
`"start"`	Content is placed to the left of the button text in LTR, and to the right in RTL.

Beispiele, die danach erklärt werden:

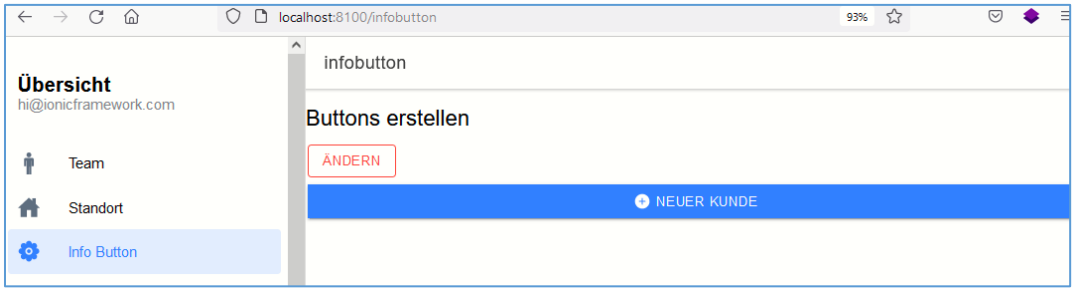
```
<ion-button expand="full" fill="outline">Outline + Full</ion-button>
<ion-button expand="block" fill="outline">Outline + Block</ion-button>
<ion-button shape="round" fill="outline">Outline + Round</ion-button>
```

Es ist egal ob steht:

- expand="full" oder
- size ="full"

Verwende „expand“:

```
9   <ion-button expand="full"><ion-icon slot="start
```




```
9   <ion-button fill="outline" color="danger">Ändern</ion-b
10  <ion-button expand="block"><ion-icon slot="start" name=
11  </ion-content>
```

Dann noch die Weite:

Expand

This attribute lets you specify how wide the button should be. By default, buttons are inline blocks, but setting this attribute will change the button to a full-width block element.

Value	Details
`block`	Full-width button with rounded corners.
`full`	Full-width button with square corners and no border on the left or right.

Und die Größe:

Size

This attribute specifies the size of the button. Setting this attribute will change the height and padding of a button.

Value	Details
`small`	Button with less height and padding. Default for buttons in an item.
`default`	Button with the default height and padding. Useful for buttons in an item.
`large`	Button with more height and padding.

Übung:

Block wieder entfernen.

CSS Layouts verwenden

LAYOUT

Structure

Responsive Grid

Global Stylesheets

CSS Utilities

z.B. für margin, padding usw.

CSS Klassen oder Attribute verwenden

Zu finden in docs und „Layout“ und „CSS Utilities“:

Info: dafür muss man keine eignen CSS-Zeilen schreiben!

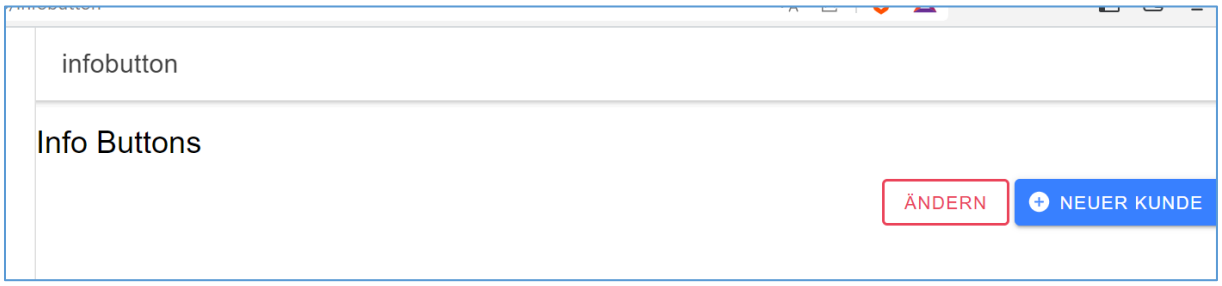
Hier findet man Attribute die man dem Element hinzufügen kann:

Class	Style Rule	Description
`.ion-text-left`	`text-align: left`	The inline contents are aligned to the left edge of the line box.
`.ion-text-right`	`text-align: right`	The inline contents are aligned to the right edge of the line box.

Übung:
Die Buttons sollen eine Margin erhalten und rechtsbündig stehen.

Dazu kann eine einfache eingefügt werden und darin eine margin und ein text-right.

```
7  <ion-content>
8    <h2>Info Buttons</h2>
9  <div class="margin ion-text-right">
10   <ion-button fill="outline" color="danger">Ändern</ion-butto
11   <ion-button><ion-icon slot="start" name="add-circle"></ion-
12 </div>
13 </ion-content>
```



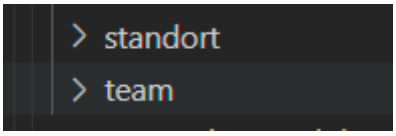
3)Weiterleitung zu einer anderen Seite

Zu einer andren Seite navigieren

3a)routerLink="page-name"

Auf welche Seite man verlinken will muss man in den bereits angelegten Seiten suchen.

z.B. team



Erstelle einen neuen Button „Zum Team"

Füge NACH dem SLASH den Namen ein:

```
10   <ion-button fill="outline" color="danger">Ändern</ion-button>
11   <ion-button><ion-icon slot="start" name="add-circle" color="warning
12   <ion-button routerLink="/team">Zum Team</ion-button>
13   </div>
```

Diese Verlinkung mittels „routerLink" funktioniert hier nur deswegen so problemlos, weil im Vorfeld schon das Router-Modul von Angular integriert hatten:

Nämlich in der „loader“page.ts“

```
1   import { Component, OnInit } from '@angular/core';
2   import { Router } from '@angular/router';
3
4   @Component({
5     selector: 'app-loader',
6     templateUrl: './loader.page.html',
7     styleUrls: ['./loader.page.scss'],
8   })
9   export class LoaderPage implements OnInit {
10
11    constructor(private router: Router) { }
```

Damit ist es auch in „app.modules.ts“ vorhanden und muss nicht extra in die „infobutton.pages.ts“ geschrieben werden.

3b)mit click-Funktion

Dafür muss man im Button eine click-Funktion erstellen, die einen eigenen Funktionsnamen erhält.

```
12    <ion-button routerLink="/team">Zum Team</ion-button>
13    <ion-button (click)="zumTeam()">Zum Team 2.Variante</ion-button>
14    </div>
```

Zusätzlich muss man in der passenden „ts-Datei“ diese neue Funktion anlegen und im constructor (bzw. import) das Router-Modul ansprechen:

```
10
11    constructor(private router: Router) { }
12
13    ngOnInit() {
14    }
15
16  zumTeam(){
17    this.router.navigate(['team']);
18  }
19
20  }
```

Und – wenn nicht automatisch improtiert – den IMPORT in Zeile 2 schreiben:

```
infobutton.page.html U ×   TS infobutton.page.ts U ×
src > app > pages > infobutton > TS infobutton.page.ts > ...
1   import { Component, OnInit } from '@angular/core';
2   import { Router } from '@angular/router';
```

4)Alert bei Click ausgeben

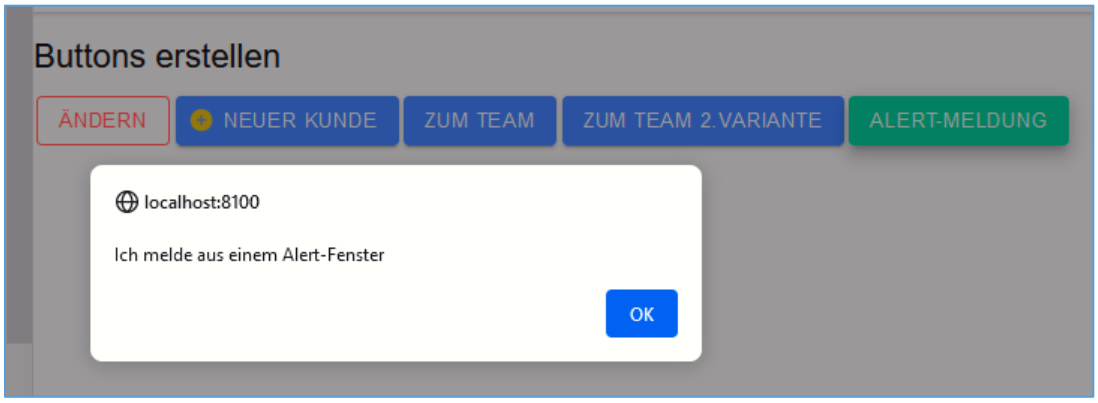
Im Button (html) wird eine click-Funktion erstellt, die auf eine alert-Funktion (in der .ts) zugreift.

```
14    <ion-button color="success" (click)="meldung()">Alert-Meldung</ion-button>
15    </div>
```

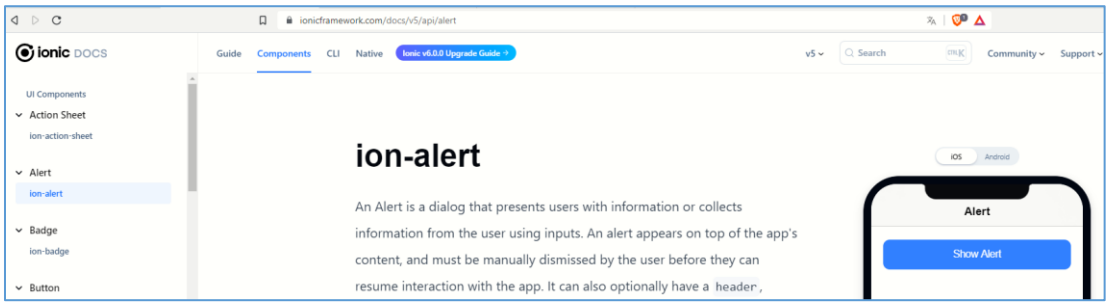
In der „infobutton.page.ts“ muss man die Funktion erstellen:

```
20  meldung(){
21    alert('Ich melde aus einem Alert-Fenster');
22  }
```

Ergebnis:



Weiteres Einsatzgebiet:

- wenn ein User sich ausloggt, kann man ihn hiermit nochmals fragen, ob er sich wirklich ausloggen will.

Infos in den „components“