

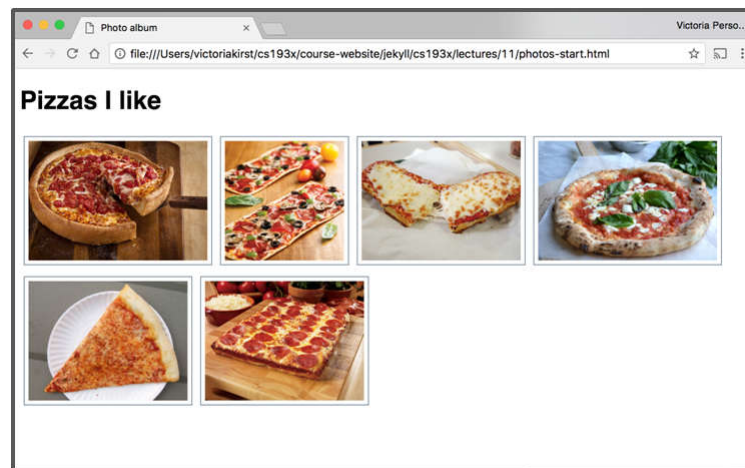
Agenda

Teme:

- Keyboard events
- Mobile events
- Jednostavne CSS animacije
- Klase i objekti
- `this` i `bind`

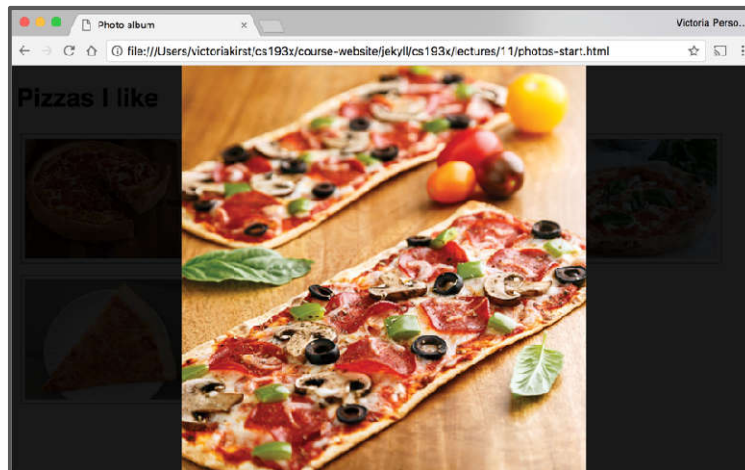
Primjer: Photo Album

Dodaćemo neke nove stvari ovom albumu :



Primjer: Photo Album

Dodaćemo neke nove stvari ovom albumu:



Pregledajmo fajlove:

[photo-start.html](#)

[photo.js](#)

[photo.css](#)

General setup

```
<body>
  <h1>Pizzas I like</h1>
  <section id="album-view">
  </section>

  <section id="modal-view" class="hidden">
  </section>
</body>
```

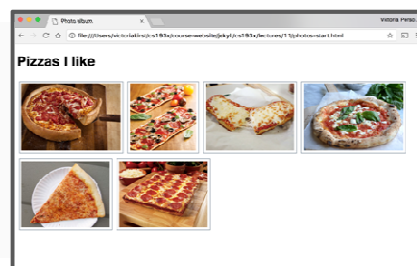
[photo.html](#) sadrži oba "ekrana:

- album view: thumbnails za svaku fotografije
- "modal" view: jedna fotografija iznad poluprozirne (semi-transparent) crne podloge – sakriven (by default)

CSS: Album

[photo.css](#): CSS za album view je jednostavan:

```
#album-view img {
  border: 1px solid slategray;
  margin: 5px;
  padding: 5px;
  height: 150px;
}
```



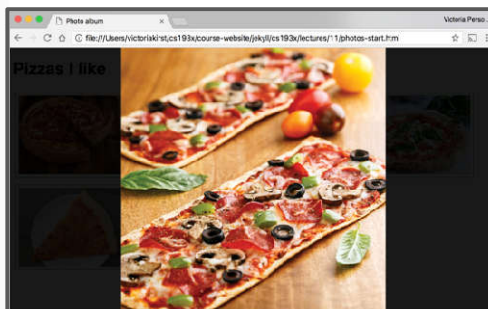
CSS: Modal

Modal view je malo složeniji:

```
#modal-view {
  position: absolute;
  top: 0;
  left: 0;
  height: 100vh;
  width: 100vw;

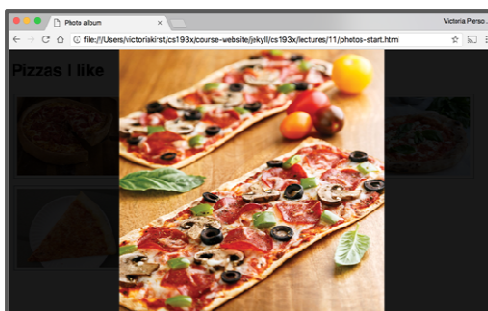
  background-color: rgba(0, 0, 0, 0.9);
  z-index: 2;

  display: flex;
  justify-content: center;
  align-items: center;
}
```



CSS: Modal image

```
#modal-view img {
  max-height: 100%;
  max-width: 100%;
}
```



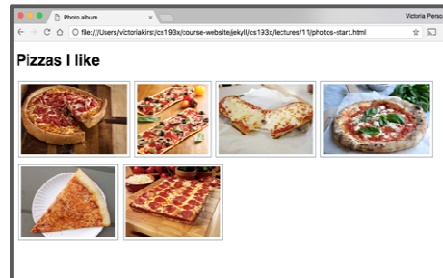
Veličina slike je ograničena na širinu i visinu roditelja (
#modal-view čija je visina i širina podešena na dimenzije
viewport)

CSS: Hidden modal

```
<body>
  <h1>Pizzas I like</h1>
  <section id="album-view">
  </section>

  <section id="modal-view" class="hidden">
  </section>
</body>
```

```
#modal-view.hidden {
  display: none;
}
```



Oba prikaza (album view i modal view) se nalaze u HTML-u, ali je model view postavljen `display: none`; pa se ne prikazuje.

Globalni niz fotografija

```
<head>
  <meta charset="utf-8">
  <title>Photo album</title>
  <link rel="stylesheet" href="css/photo.css">
  <script src="js/photo-list.js" defer></script>
  <script src="js/photo.js" defer></script>
</head>
```

```
const PHOTO_LIST = [
  'images/deepdish.jpg',
  'images/flatbread.jpg',
  'images/frenchbread.jpg',
  'images/neapolitan.jpg',
  'images/nypizza.jpg',
  'images/squarepan.jpg'
];
```

[photo-list.js](#): globalni niz `PHOTO_LIST` koji sadrži stringove koji su imena fotografija (putanje do fotografija).

Photo thumbnails

```
function createImage(src) {
  const image = document.createElement('img');
  image.src = src;
  return image;
}

const albumView = document.querySelector('#album-view');
for (let i = 0; i < PHOTO_LIST.length; i++) {
  const photoSrc = PHOTO_LIST[i];
  const image = createImage(photoSrc);
  image.addEventListener('click', onThumbnailClick);
  albumView.appendChild(image);
}
```

[photo.js](#): Popunjavanje početnog albuma: ciklus po nizu PHOTO_LIST i nadovezivanje na #album-view.

Klik na sliku

```
function onThumbnailClick(event) {
  const image = createImage(event.currentTarget.src);
  modalView.appendChild(image);
  modalView.classList.remove('hidden');
}
```

Kada korisnik klikne na sliku:

- Kreiramo novi tag sa istim atributom src
- Nadovežemo novi na #modal-view
- Postavimo da je #modal-view vidljiv

Pozicioniranje modalnog pogleda

```
function onThumbnailClick(event) {
  const image = createImage(event.currentTarget.src);
  modalView.style.top = window.pageYOffset + 'px';
  modalView.appendChild(image);
  modalView.classList.remove('hidden');
}
```

Dodajemo liniju koda u JavaScript da i utvrdili poziciju na “top of the viewport” ne za “top of the screen”:

```
modalView.style.top = window.pageYOffset + 'px';
```

(Pogledajte [window.pageYOffset mdn](#). Ili [window.scrollTo](#).)

Atribut style


Svakom [HTMLElement](#) može se doadti atribut [style](#) koji zadaje stil tog elemenata:

```
element.style.top = window.pageYOffset + 'px';
```

U opštem slučaju **ne treba koristiti ovo svojstvo**, jer je dodavanje i uklanjanje klasa pomoću [classList](#) bolji način za promjenu stila elementa

Ali ako mijenjamo CSS svojstvo na osnovu vrijednosti iz JavaScript-A, moramo postaviti atribut `style` neposredno .

No scroll na stranici



```
function onThumbnailClick(event) {
  const image = createImage(event.currentTarget.src);
  document.body.classList.add('no-scroll');
  modalView.style.top = window.pageYOffset + 'px';
  modalView.appendChild(image);
  modalView.classList.remove('hidden');
}

.no-scroll {
  overflow: hidden;
}
```

Zabranjujemo skrolovanje na stranici tako što postavimo `body { overflow: hidden; }` – ovo je jedan način da se to uradi.

Zatvaranje modalnog dijaloga

```
function onModalClick() {
  document.body.classList.remove('no-scroll');
  modalView.classList.add('hidden');
  modalView.innerHTML = '';
}

const modalView = document.querySelector('#modal-view');
modalView.addEventListener('click', onModalClick);
```

Kad korisnik klikne na modal view:

- Ponovo sakrivamo modal view
- Dopuštamo skrolovanje na stranici
- Uklanjamo veliku sliku pomoću `innerHTML = ''`;