

Въведение в Angular

Минко Гечев, май 2022

ФМИ, 2010

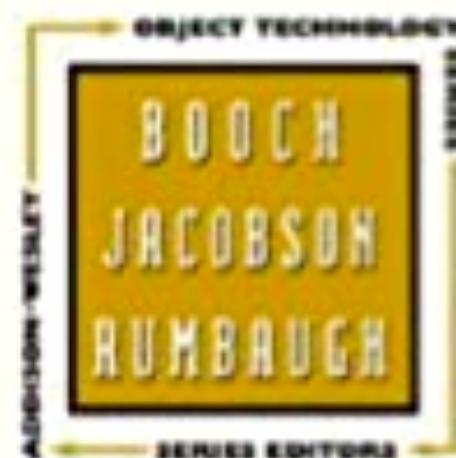




OBJECT-ORIENTED ANALYSIS AND DESIGN WITH APPLICATIONS

THIRD EDITION

GRADY BOOCH, ROBERT A. MAKSIMCHUK,
MICHAEL W. ENGLE, BOBBI J. YOUNG, Ph.D.,
JIM CONALLEN, KELLI A. HOUSTON



```
var newstr = null;
  var lastnewstr = null;
  var newclass = null;
  ...

function check(s, i) {
  p1 = document.getElementById('id_1_' + i);
  p2 = document.getElementById('id_2_' + i);
  if (s !== lastnewstr)
  {
    newclass2 = p1.className;
    newclass3 = p2.className;
    newclass = s.className;
    s.className = 'check';
    p1.className = 'viz';
    p2.className = 'viz';
    if (lastnewstr !== null)
    {
      lastnewstr.className = lastclass;
      lastnewstr2.className = lastclass2;
      lastnewstr3.className = lastclass3;
    }
  }
  lastnewstr = s;
  lastnewstr2 = p1;
  lastnewstr3 = p2;
  lastclass = newclass;
  lastclass2 = newclass2;
  lastclass3 = newclass3;
}
```







Angular ме научи на:

- Шаблони за дизайн
- Писане на тестваем код
- Разделяне на отговорностите
- Много други

Какво е Angular?



The modern web developer's platform

[GET STARTED](#)



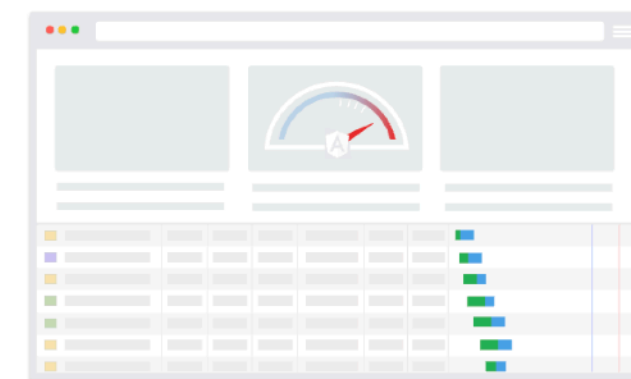
DEVELOP ACROSS ALL PLATFORMS

Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.

SPEED & PERFORMANCE

Achieve the maximum speed possible on the Web Platform today, and take it further, via Web Workers and server-side rendering.

Angular puts you in control over scalability. Meet huge data requirements by building data models on RxJS, Immutable.js or another push-model.





The modern web developer's platform

[GET STARTED](#)



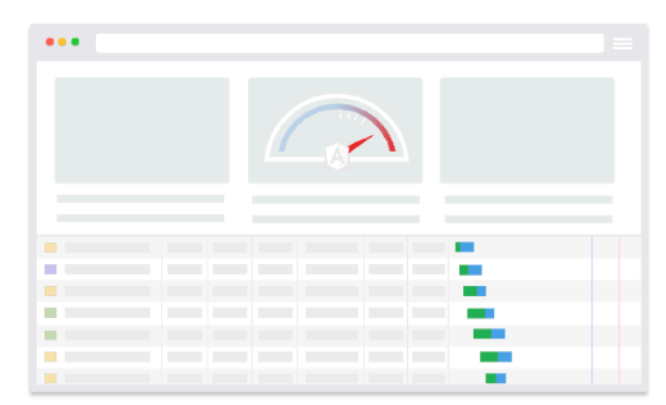
DEVELOP ACROSS ALL PLATFORMS

Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.

SPEED & PERFORMANCE

Achieve the maximum speed possible on the Web Platform today, and take it further, via Web Workers and server-side rendering.

Angular puts you in control over scalability. Meet huge data requirements by building data models on RxJS, Immutable.js or another push-model.





The **modern** web developer's platform

[GET STARTED](#)



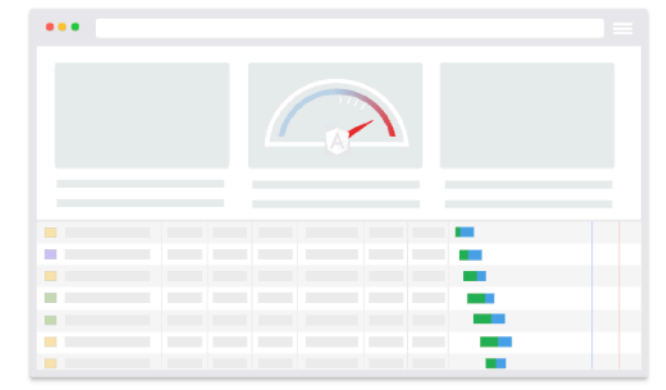
DEVELOP ACROSS ALL PLATFORMS

Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.

SPEED & PERFORMANCE

Achieve the maximum speed possible on the Web Platform today, and take it further, via Web Workers and server-side rendering.

Angular puts you in control over scalability. Meet huge data requirements by building data models on RxJS, Immutable.js or another push-model.



Цели на Angular

- Улеснява създаването на уеб приложения
- Инструменти, които да подържат процеса ви на работа
- Добри практики от самото начало
- Инкрементално въведение на концепции
- Предвиден за скалируемост
- Автоматично (лесно) обновяване към нови версии

ОСНОВНИ ТЕХНОЛОГИИ



В зависимост от нуждите ви:

- Прimitives за разработка на приложения ([@angular/cdk](#))
- Рутер ([@angular/router](#))
- Форми ([@angular/forms](#))
- Компоненти ([@angular/material](#))
- Progressive Web Apps ([@angular/pwa](#))
- Рендериране на сървъра ([@nguniversal/express-engine](#))
- ...

Не е нужно да знаете **ВСИЧКИ** тези
технологии, за да сте продуктивни

Концептуално въведение в Angular

- Компоненти (@Component)
- Модули* (@NgModule)

Допълнителни концепции

...за по-добро разделяне на отговорностите

- Услуги (@Injectable)
- Форматиране (@Pipe)
- Директиви (@Directive)

Не е нужно да ги използвате от самото начало

Hello, world

```
@Component({  
  selector: 'app-root',  
  template: 'Hello, world!'  
})  
export class AppComponent {}
```

Компоненти

- Имат селектор, който се използва за идентификацията им
- Имат темплейт и стилове, които се използват за рендериране на изгледа
- Могат да бъдат параметризирани (@Input)
- Могат да излъчват събития (@Output)

```

@Component({
  selector: 'grocery-list',
  template: `
    <grocery
      *ngFor="let g of groceries"
      [title]="g" (delete)="delete($event)">
    </grocery>
  `
})
export class GroceryList {
  groceries = ['bananas', 'milk'];

  delete(item: string) {
    this.groceries.splice(this.groceries.indexOf(item), 1);
  }
}

@Component({
  selector: 'grocery',
  template: '{{title}} <button (click)="delete.emit(title)">Delete</button>'
})
export class Grocery {
  @Input() title = '';
  @Output() delete = new EventEmitter<string>();
}

```

```
@Component({
  selector: 'grocery-list',
  template: `
    <grocery
      *ngFor="let g of groceries"
      [title]="g" (delete)="delete($event)">
    </grocery>
  `
})
export class GroceryList {
  groceries = ['bananas', 'milk'];

  delete(item: string) {
    this.groceries.splice(this.groceries.indexOf(item), 1);
  }
}
```

```
@Component({
  selector: 'grocery',
  template: '{{title}} <button (click)="delete.emit(title)">Delete</button>'
})
export class Grocery {
  @Input() title = '';
  @Output() delete = new EventEmitter<string>();
}
```

```
@Component({
  selector: 'grocery-list',
  template: `
    <grocery
      *ngFor="let g of groceries"
      [title]="g" (delete)="delete($event)">
    </grocery>
  `
})
export class GroceryList {
  groceries = ['bananas', 'milk'];

  delete(item: string) {
    this.groceries.splice(this.groceries.indexOf(item), 1);
  }
}
```

```
@Component({
  selector: 'grocery',
  template: '{{title}} <button (click)="delete.emit(title)">Delete</button>'
})
export class Grocery {
  @Input() title = '';
  @Output() delete = new EventEmitter<string>();
}
```

Angular модули

Angular модули

```
@NgModule({  
  declarations: [Grocery, GroceryList]  
})  
export class GroceryModule {}
```


Angular модули

```
@Component({
  selector: 'grocery-list',
  template: `
    <grocery
      *ngFor="let g of groceries"
      [title]="g" (delete)="delete($event)">
    </grocery>
  `
})
export class GroceryList { ... }
```

```
@Component({
  selector: 'grocery',
  template: ' ... '
})
export class Grocery { ... }
```

Dependency injection

Dependency injection

```
@Injectable({
  providedIn: 'root'
})
export class Employee {
  delete(id: string): Promise<void> {
    return fetch( ... );
  }
}

@Component({
  selector: 'app-root',
  template: '<button (click)="delete()">Delete</button>'
})
export class AppComponent {
  constructor(private service: Employee) {}
  delete() {
    this.service.delete( ... );
  }
}
```

Обобщение

- За създаване на приложения използвайте Angular CLI
- Приложенията са композиция от компоненти
- Компонентите имат:
 - Селектор
 - Темплейт и стилове
 - Параметри (@Input)
 - Събития (@Output)
- Декларирайте бизнес логика в услуги
- Модулите определят контекста на компонент

И едно последно нещо...

@angular/router

- Позволява навигация без презереждане на страницата
- Можете да параметризирате и влагате маршрути
- Децентрализирана дефиниция на маршрути

Дефиниция на маршрути

```
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    RouterModule.forRoot([
      {
        path: '',
        pathMatch: 'full',
        component: HomeComponent
      },
      {
        path: ':user_id',
        loadChildren: import('./user/user.module').then(m => m.UserModule)
      }
    ])
  ],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

**Къде ще бъдат рендерирани
маршрутите?**

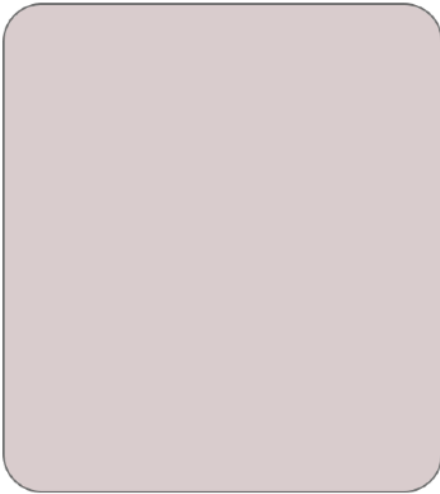

```
@Component({
  selector: 'app-root',
  template: `
    <nav>Navigation</nav>
    <router-outlet></router-outlet>
    <footer>Footer</footer>
  `
})
export class AppComponent {}
```

**За повече
информация**
angular.io/start

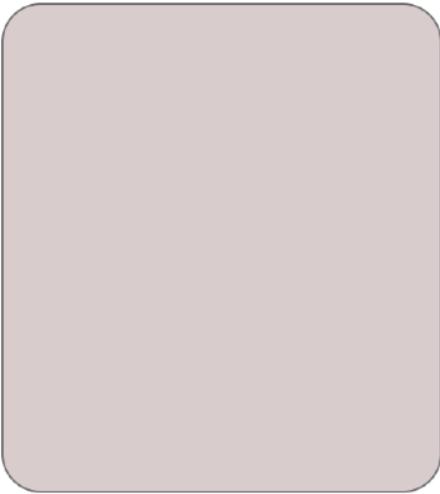


Демонстрация

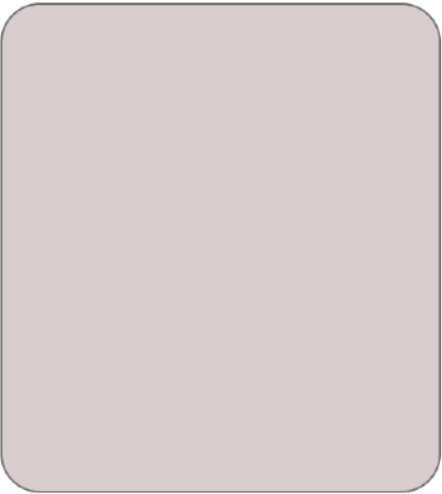
Gallery Search



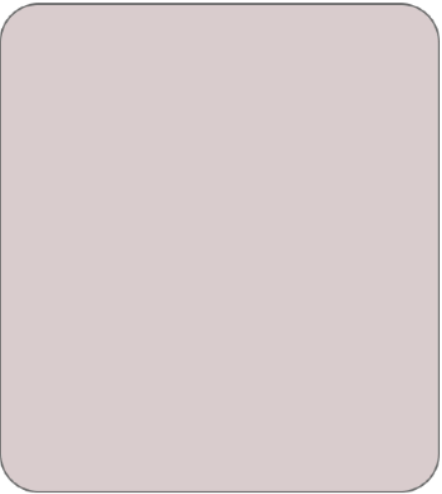
Firstname
Lastname



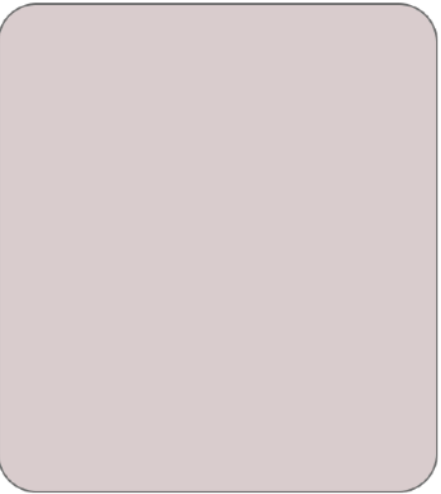
Firstname
Lastname



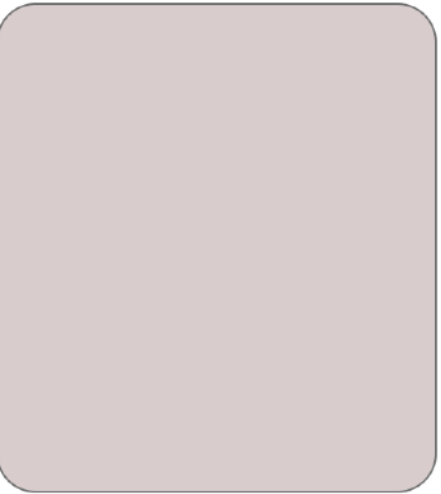
Firstname
Lastname



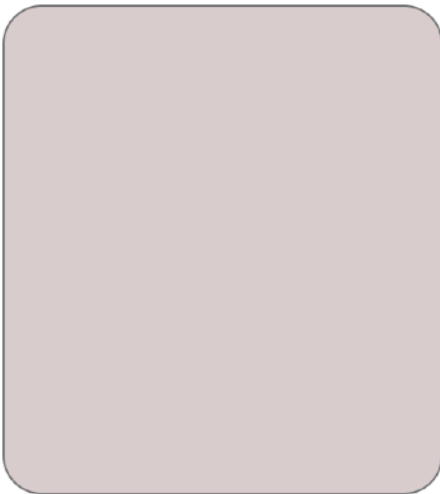
Firstname
Lastname



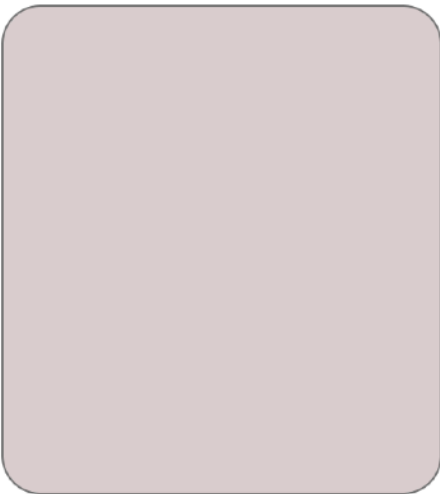
Firstname
Lastname



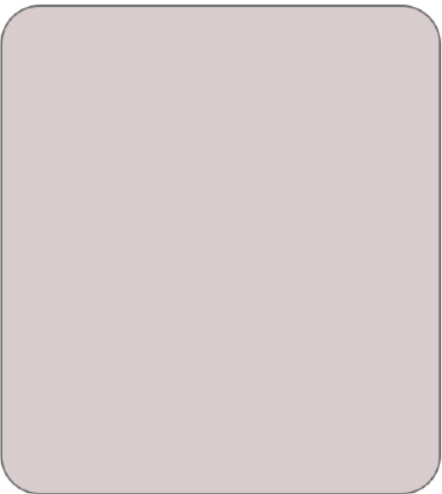
Firstname
Lastname



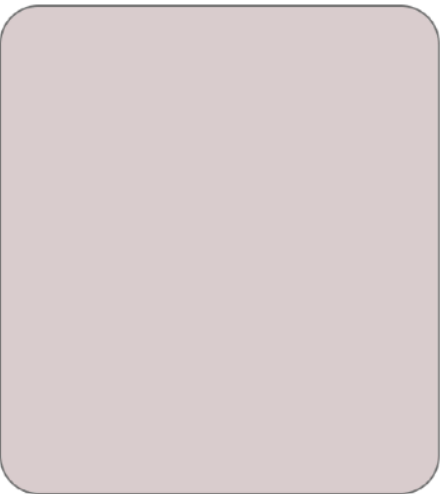
Firstname
Lastname



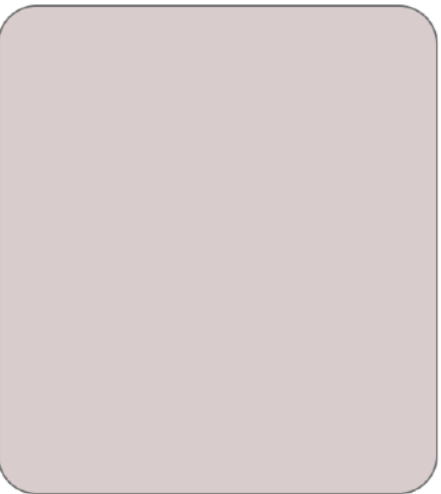
Firstname
Lastname



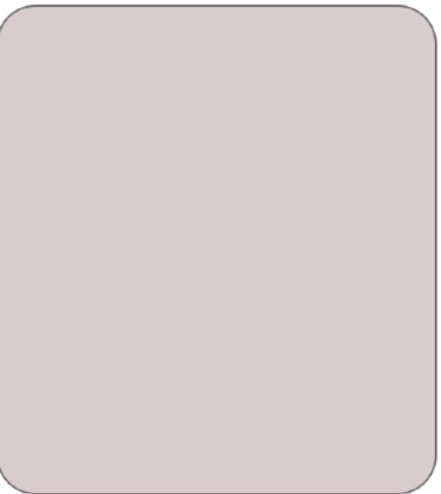
Firstname
Lastname



Firstname
Lastname

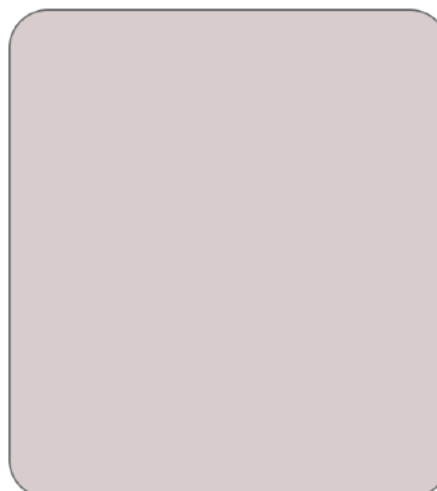


Firstname
Lastname



Firstname
Lastname

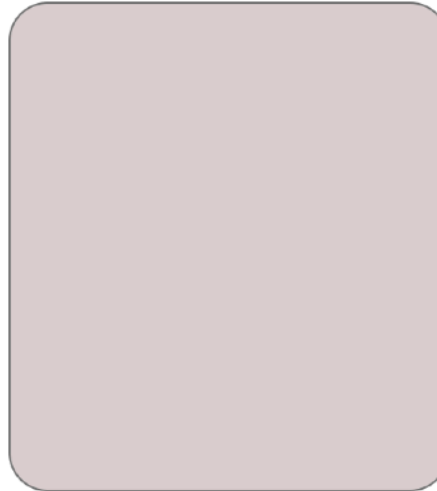
Gallery Search



Firstname
Lastname



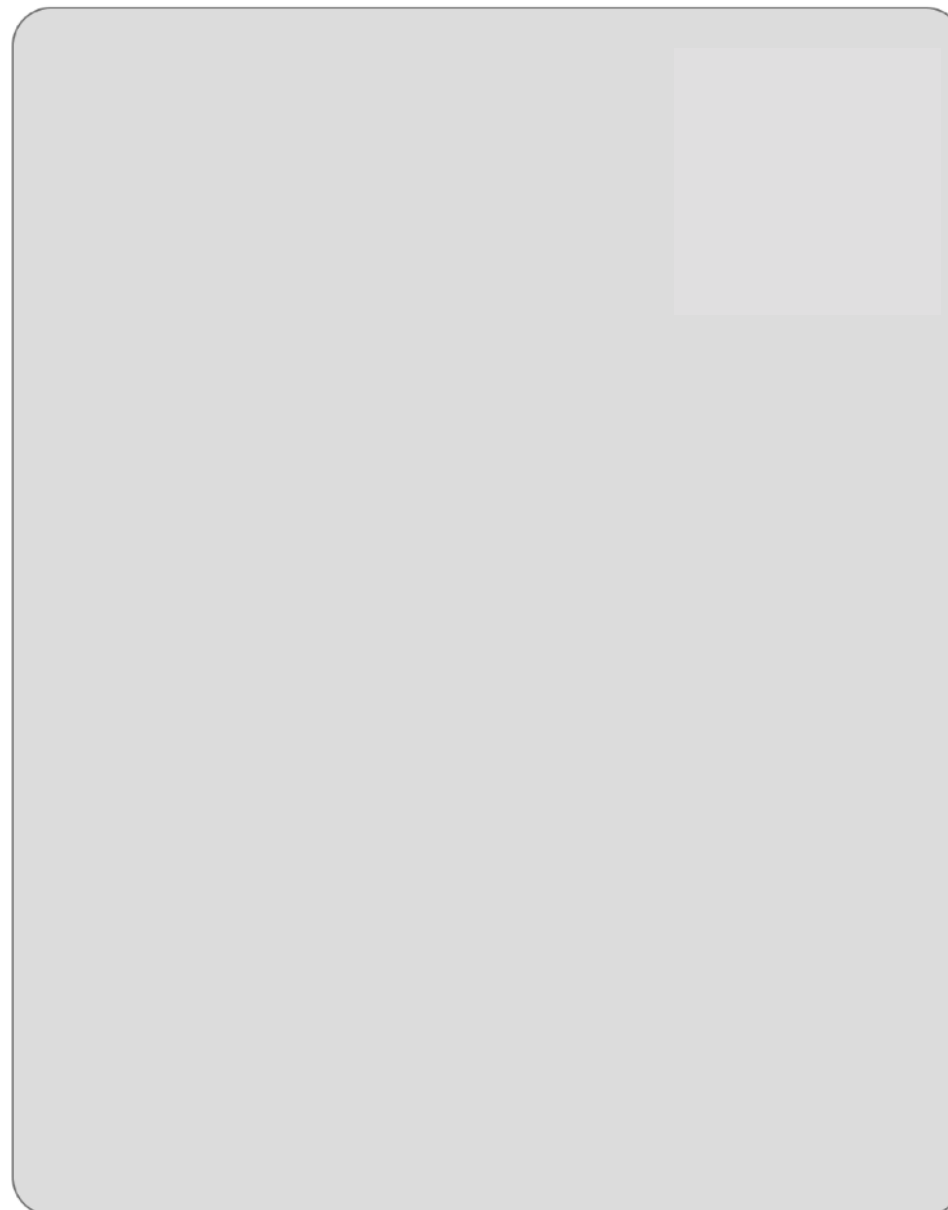
Fi
La



Firstname
Lastname



Fi
La



Firstname Lastname 

I study at Sofia University's Faculty of Mathematics and Informatics. My favorite programming language is TypeScript. 

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Благодаря за вниманието :-)