cześć

cześć

Should we update apps to #angular 17?

Przemysław Sokołowski

cześć

Przemysław Sokołowski

Synergy Codes | Fullstack dev

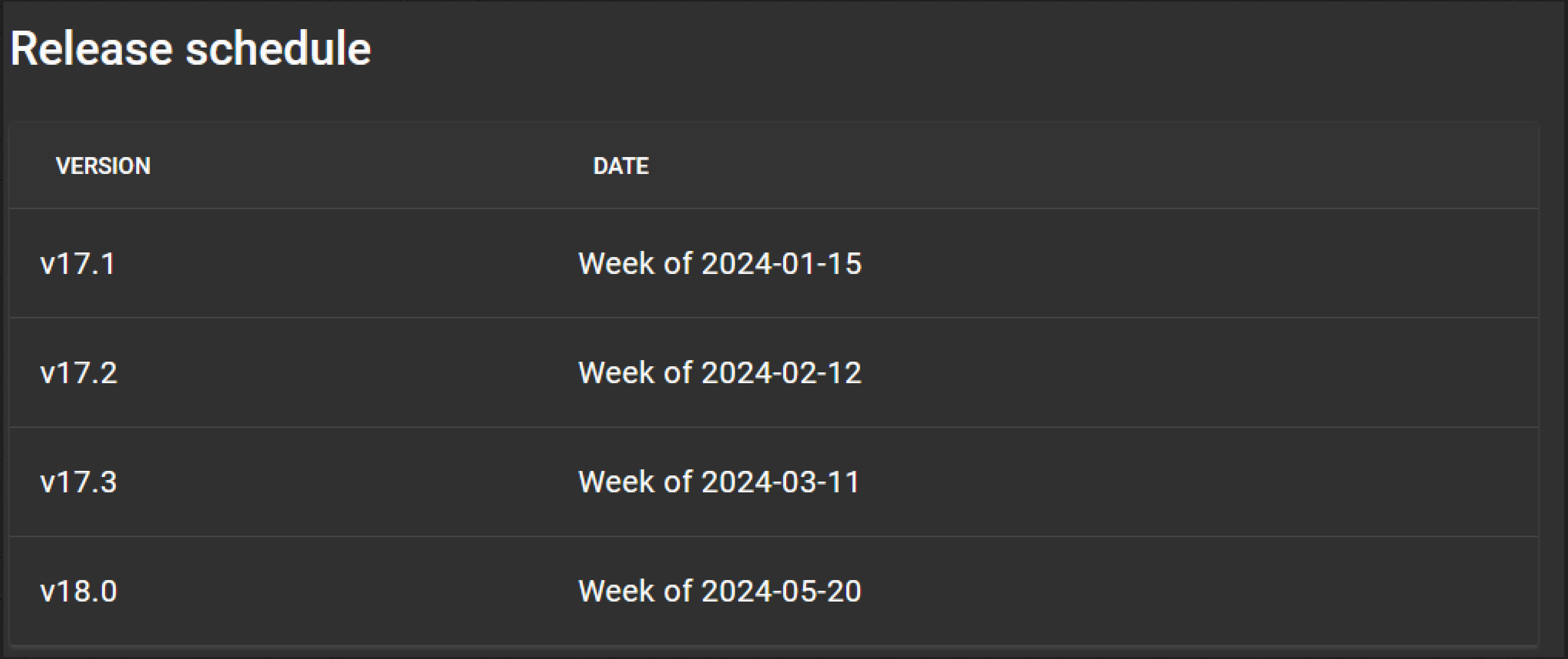
Github: bboysokol

LinkedIn: przemysław-sokołowski

Agenda

- Harmonogram wydań
- Built-in control flow
- Branding
- Dokumentacja
- Deferrable views
- Styles and style
- SSR and Hydration
- Signals
- New router info parameter
- TypeScript support
- ESBuild and Vite

Harmonogram wydań



Release schedule

VERSION	DATE
v17.1	Week of 2024-01-15
v17.2	Week of 2024-02-12
v17.3	Week of 2024-03-11
v18.0	Week of 2024-05-20

Źródło: https://angular.io/guide/releases#release-schedule

Built-in control flow

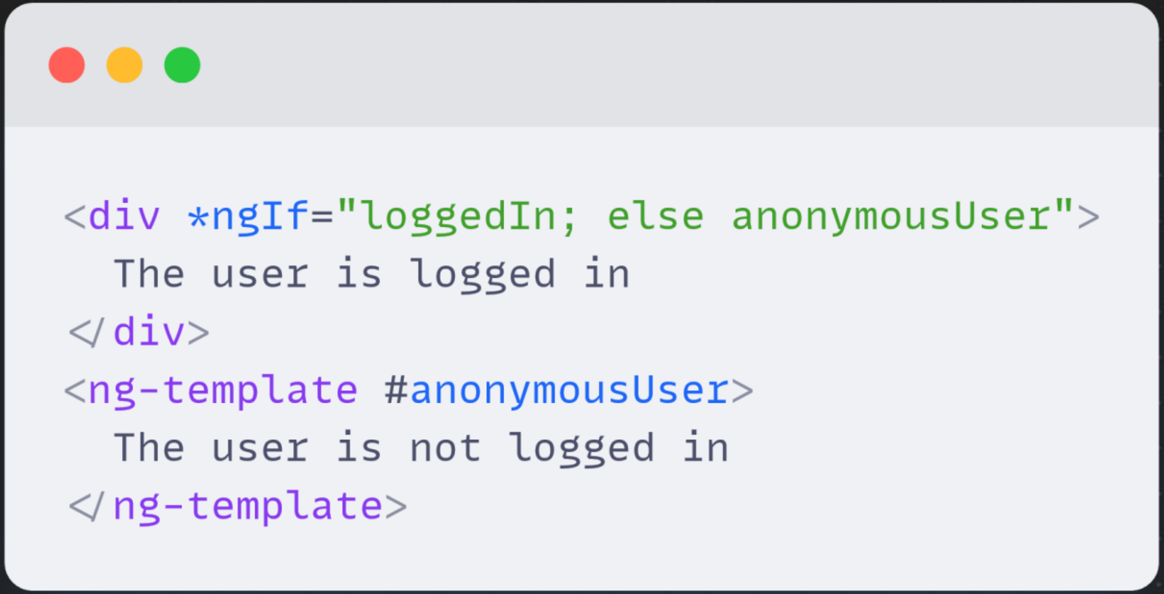
Built-in control flow

wprowadzenie

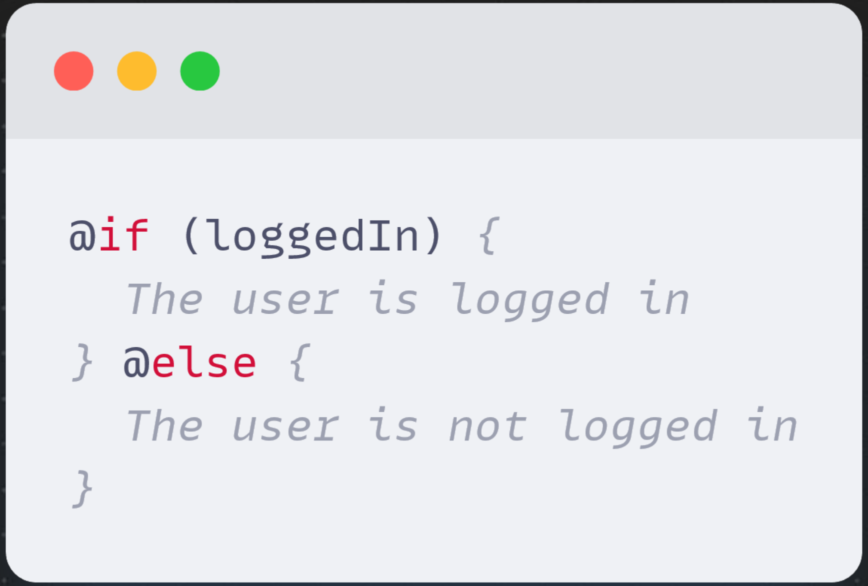
- Niższy próg wejścia
- Lepsze sprawdzanie typów
- Mniejszy bundle
- Jest dostępny domyślnie
- Znacząca poprawa wydajności

Built-in control flow

*ngIf



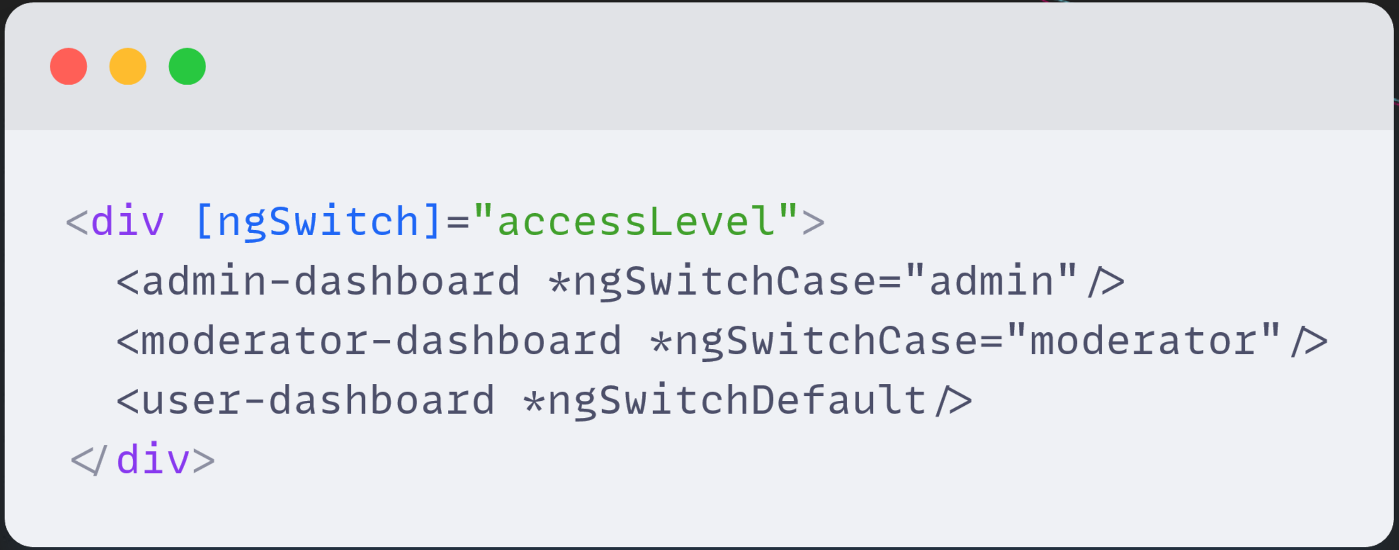
```
<div *ngIf="loggedIn; else anonymousUser">
  The user is logged in
</div>
<ng-template #anonymousUser>
  The user is not logged in
</ng-template>
```



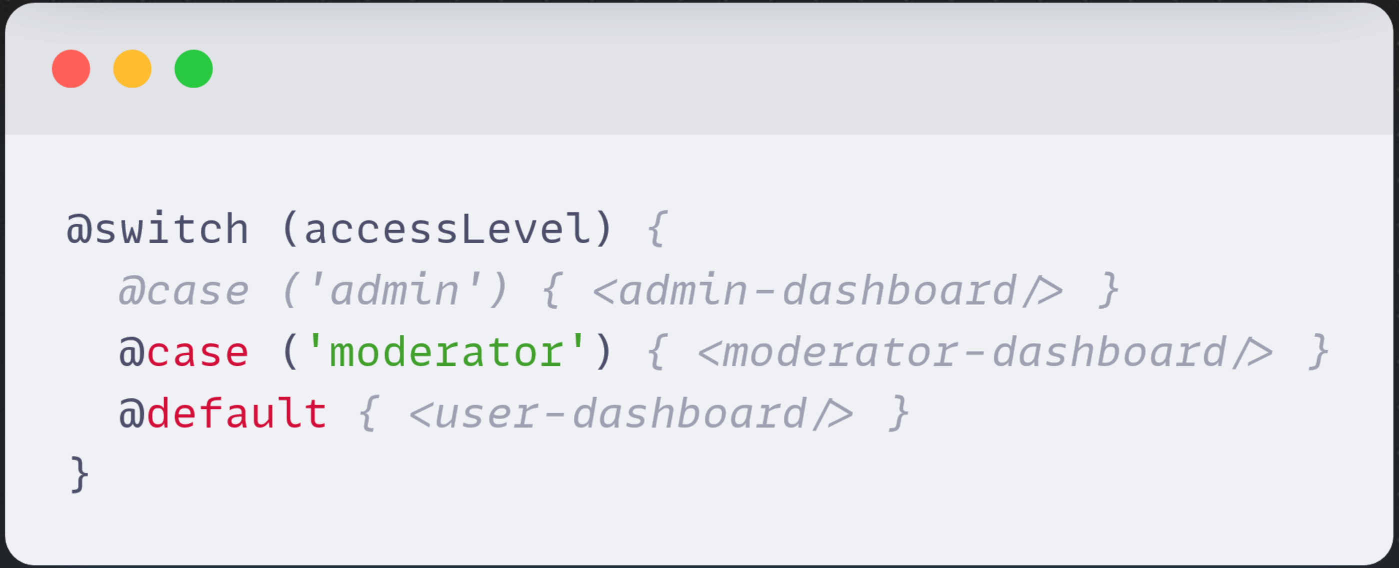
```
@if (loggedIn) {
  The user is logged in
} @else {
  The user is not logged in
}
```

Built-in control flow

*ngSwitch



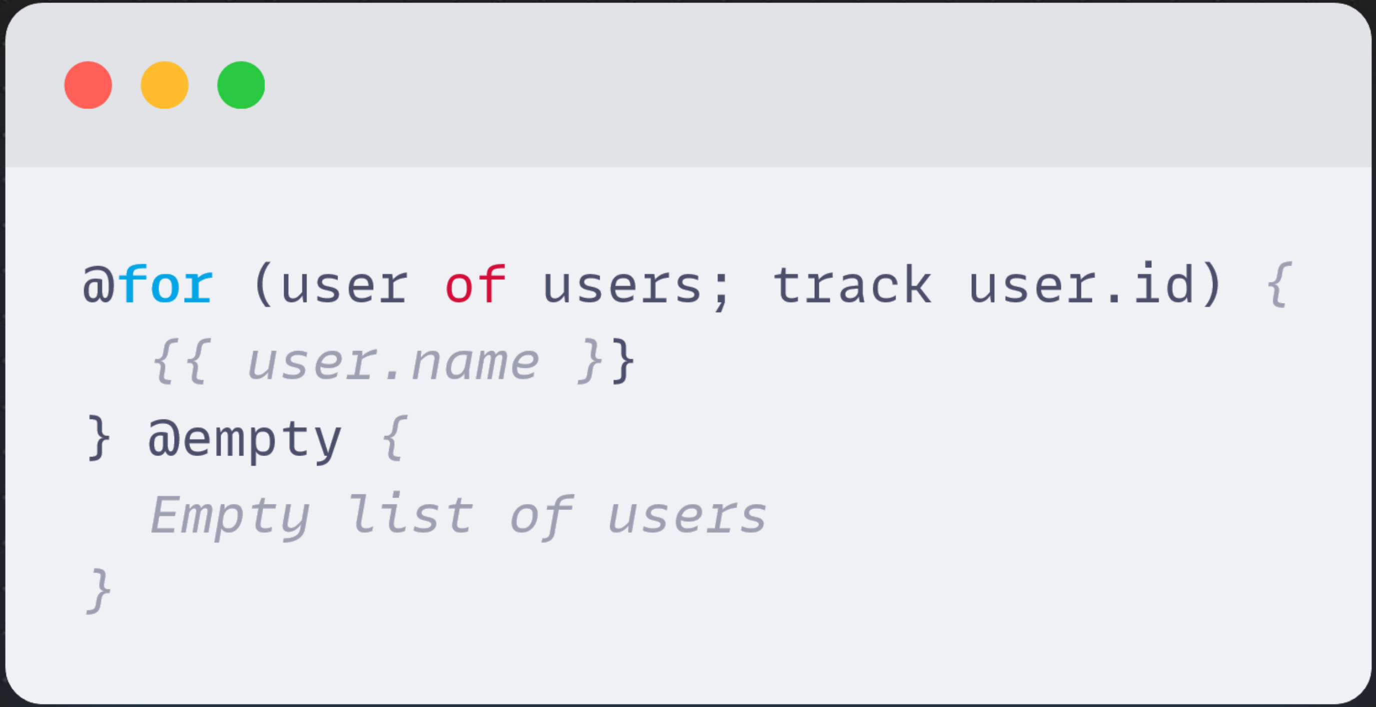
```
<div [ngSwitch]="accessLevel">
  <admin-dashboard *ngSwitchCase="admin"/>
  <moderator-dashboard *ngSwitchCase="moderator"/>
  <user-dashboard *ngSwitchDefault/>
</div>
```



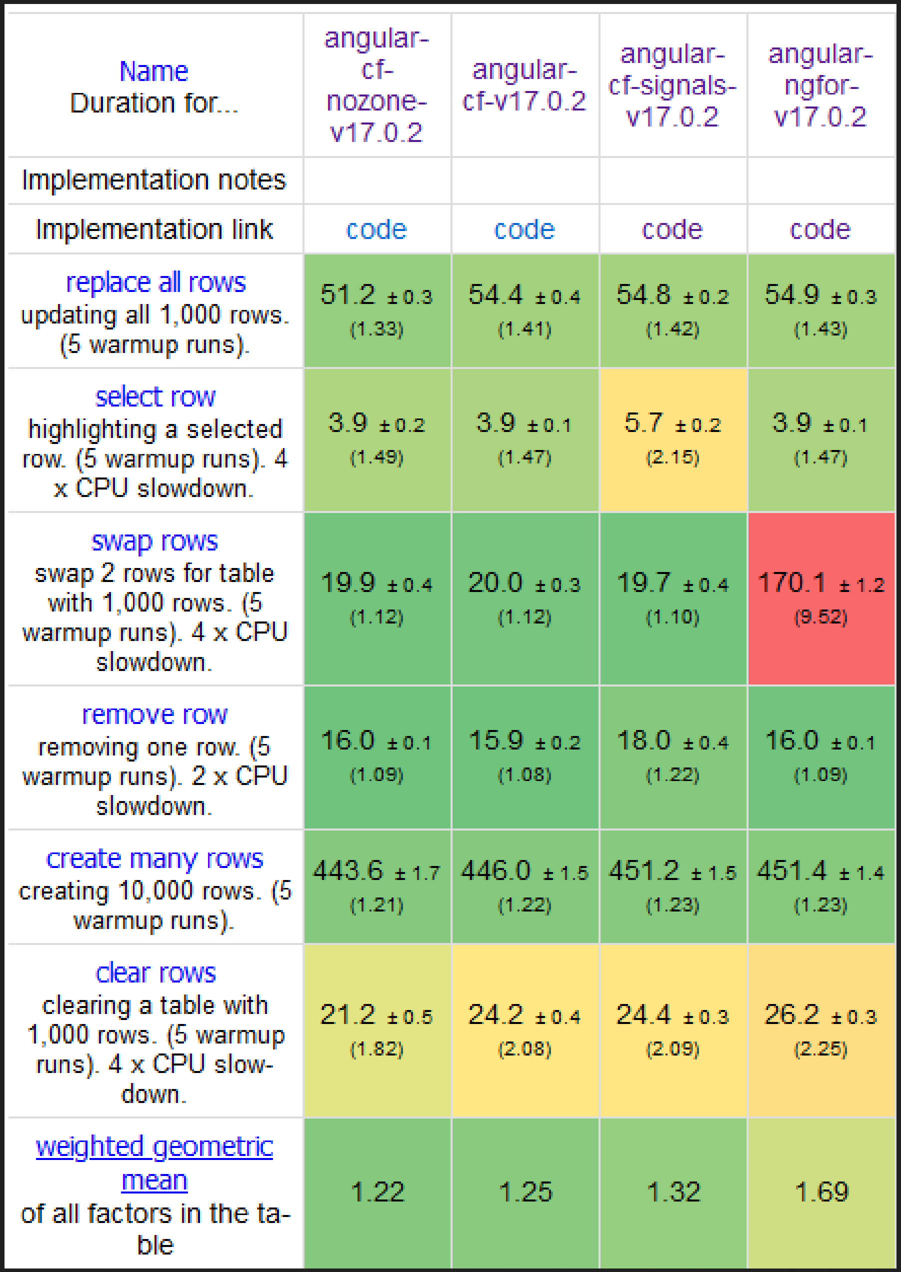
```
@switch (accessLevel) {
  @case ('admin') { <admin-dashboard/> }
  @case ('moderator') { <moderator-dashboard/> }
  @default { <user-dashboard/> }
}
```

Built-in control flow

*ngFor



```
@for (user of users; track user.id) {
  {{ user.name }}
} @empty {
  Empty list of users
}
```

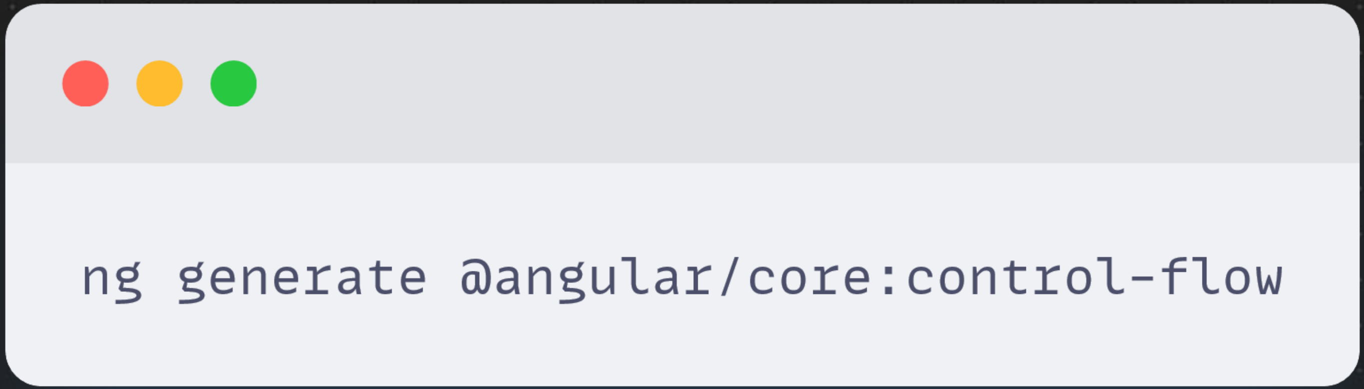


Name Duration for...	angular-cf-nozone-v17.0.2	angular-cf-v17.0.2	angular-cf-signals-v17.0.2	angular-ngfor-v17.0.2
Implementation notes				
Implementation link	code	code	code	code
replace all rows updating all 1,000 rows. (5 warmup runs).	51.2 ± 0.3 (1.33)	54.4 ± 0.4 (1.41)	54.8 ± 0.2 (1.42)	54.9 ± 0.3 (1.43)
select row highlighting a selected row. (5 warmup runs). 4 x CPU slowdown.	3.9 ± 0.2 (1.49)	3.9 ± 0.1 (1.47)	5.7 ± 0.2 (2.15)	3.9 ± 0.1 (1.47)
swap rows swap 2 rows for table with 1,000 rows. (5 warmup runs). 4 x CPU slowdown.	19.9 ± 0.4 (1.12)	20.0 ± 0.3 (1.12)	19.7 ± 0.4 (1.10)	170.1 ± 1.2 (9.52)
remove row removing one row. (5 warmup runs). 2 x CPU slowdown.	16.0 ± 0.1 (1.09)	15.9 ± 0.2 (1.08)	18.0 ± 0.4 (1.22)	16.0 ± 0.1 (1.09)
create many rows creating 10,000 rows. (5 warmup runs).	443.6 ± 1.7 (1.21)	446.0 ± 1.5 (1.22)	451.2 ± 1.5 (1.23)	451.4 ± 1.4 (1.23)
clear rows clearing a table with 1,000 rows. (5 warmup runs). 4 x CPU slowdown.	21.2 ± 0.5 (1.82)	24.2 ± 0.4 (2.08)	24.4 ± 0.3 (2.09)	26.2 ± 0.3 (2.25)
weighted geometric mean of all factors in the table	1.22	1.25	1.32	1.69

Źródło: https://krausest.github.io/js-framework-benchmark/current.html

Built-in control flow

migracja

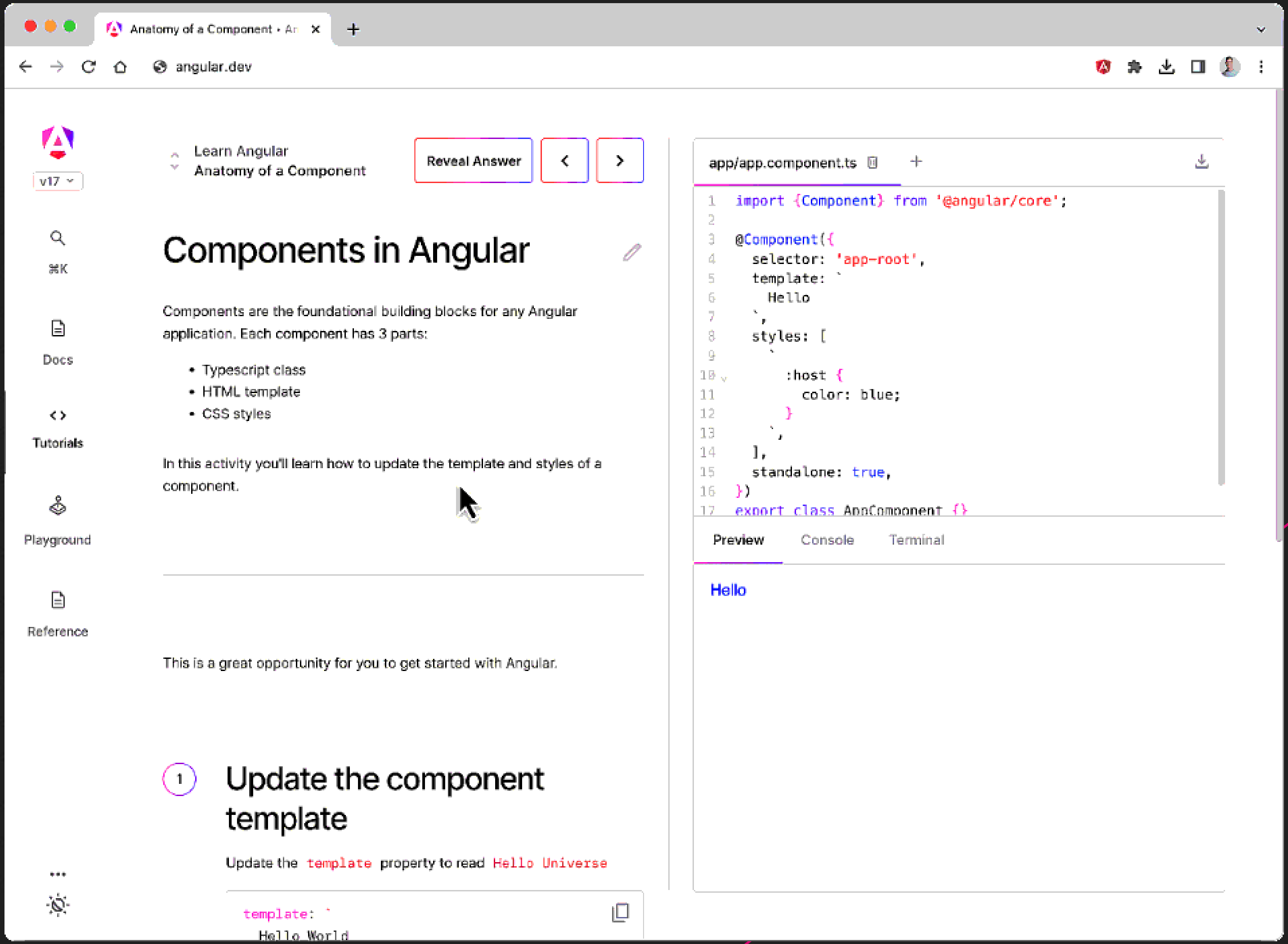


```
ng generate @angular/core:control-flow
```

DIRECTIVES
HTML
ANGULAR CONTROL FLOW SYNTAX
What the hell is this?
imgflip.com

Branding

Dokumentacja



Learn Angular
Anatomy of a Component

Components in Angular

Components are the foundational building blocks for any Angular application. Each component has 3 parts:

- Typescript class
- HTML template
- CSS styles

In this activity you'll learn how to update the template and styles of a component.

This is a great opportunity for you to get started with Angular.

1 Update the component template

Update the `template` property to read `Hello Universe`

```
template: `
  Hello World
```

```
import {Component} from '@angular/core';

@Component({
  selector: 'app-root',
  template: `
    Hello
  `,
  styles: [
    `
      :host {
        color: blue;
      }
    `,
  ],
  standalone: true,
})
export class AppComponent {}
```

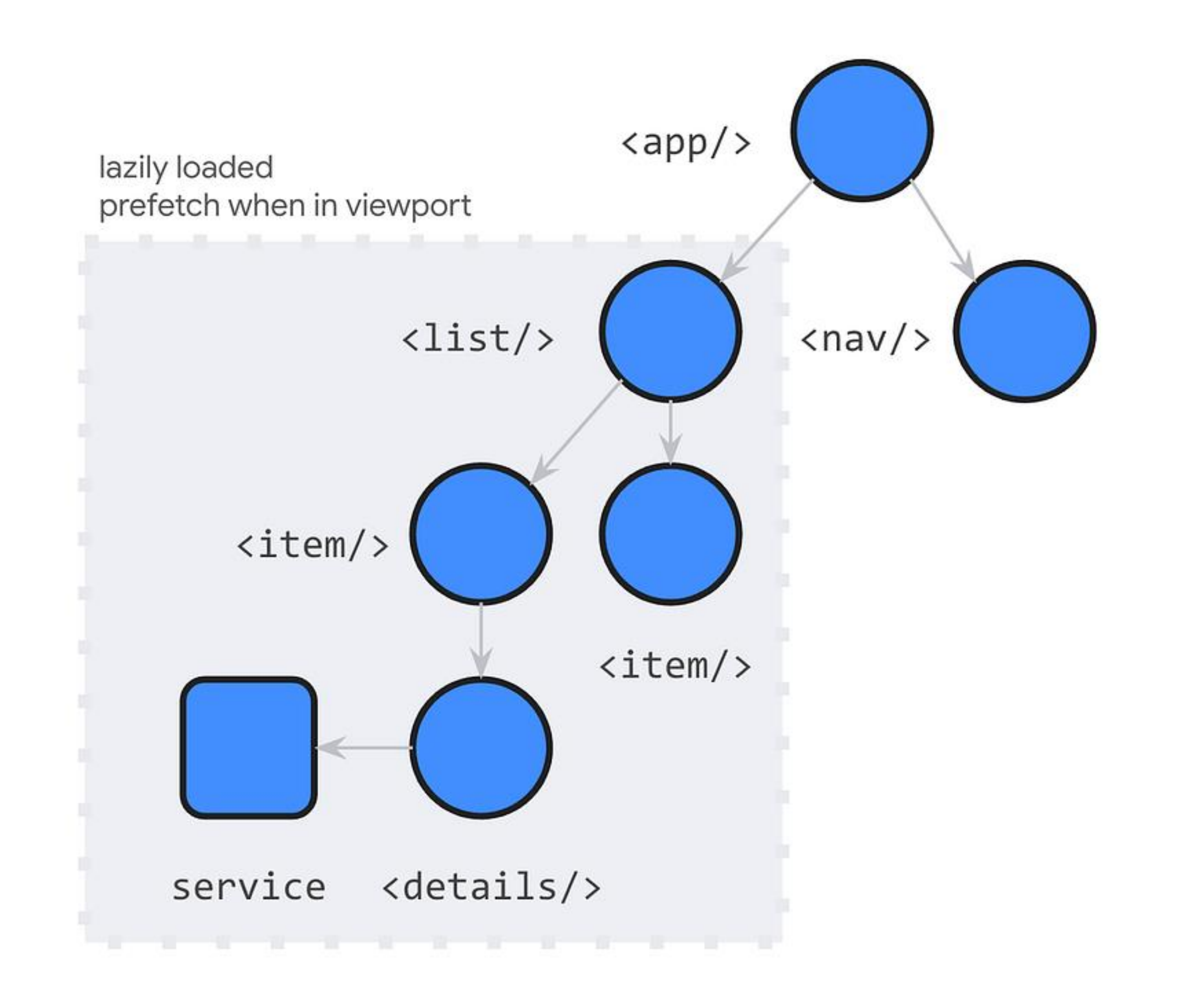
Hello

Deferrable views

Deferrable views

wprowadzenie

- Lazy loading przy użyciu deklaratywnej składni



Źródło: https://blog.angular.io/introducing-angular-v17-4d7033312e4b

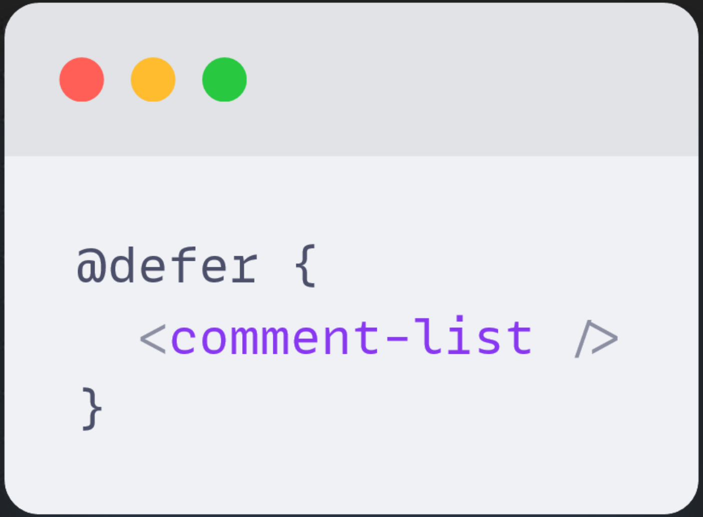
Deferrable views

triggery

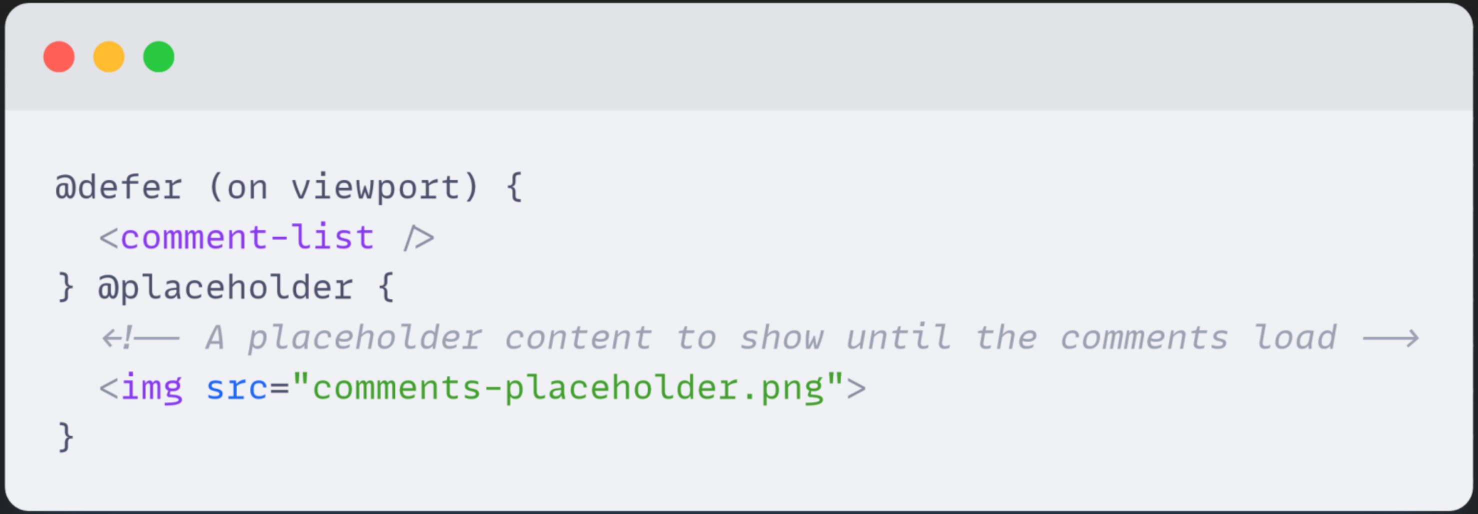
- on idle
- on immediate
- on timer(<time>)
- on viewport and on viewport(<ref>)
- on interaction and on interaction(<ref>)
- on hover and on hover(<ref>)
- when <expr>

Deferrable views

praktyka

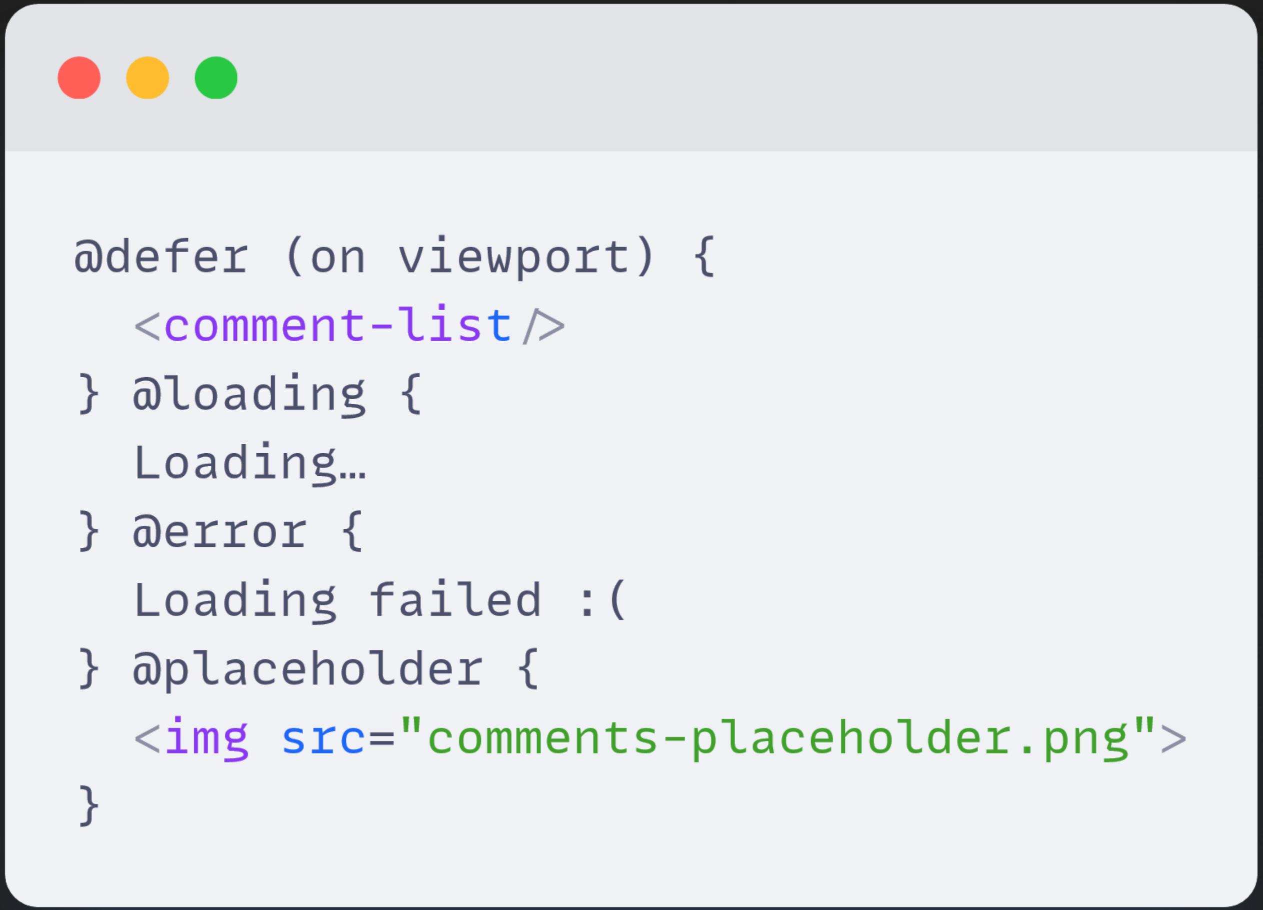


```
@defer {
  <comment-list />
}
```



```
@defer (on viewport) {
  <comment-list />
} @placeholder {
  <!-- A placeholder content to show until the comments load -->
  <img src="comments-placeholder.png">
}
```

Przykład z: https://blog.angular.io/introducing-angular-v17-4d7033312e4b

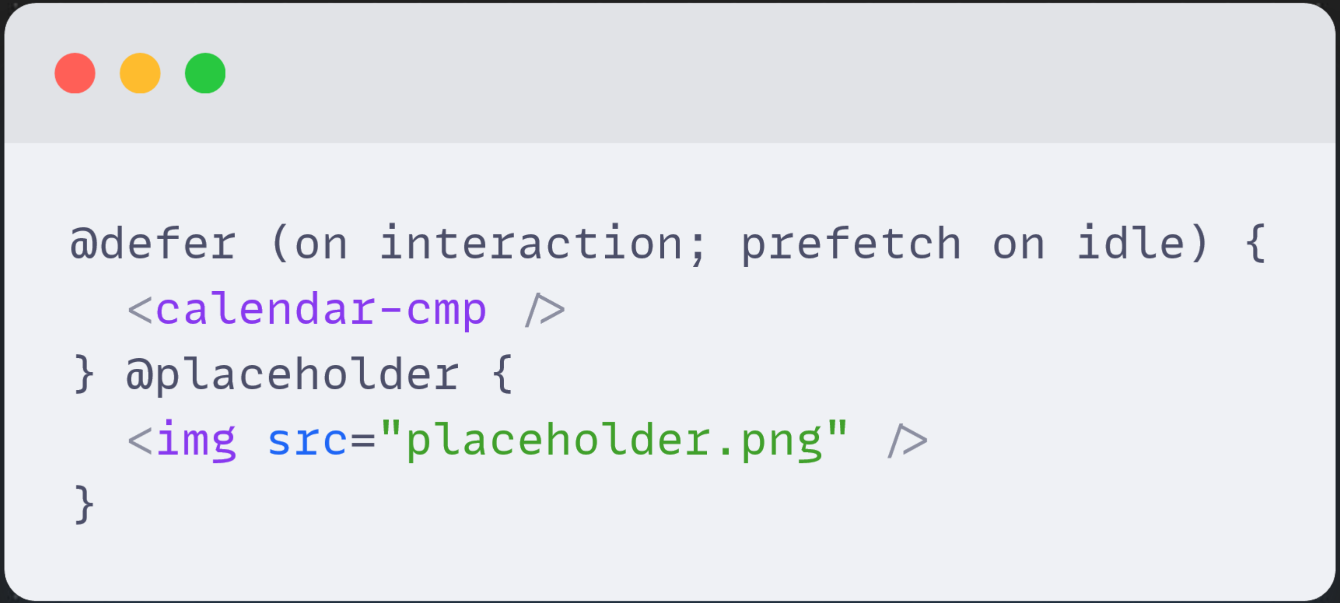


```
@defer (on viewport) {
  <comment-list/>
} @loading {
  Loading…
} @error {
  Loading failed :(
} @placeholder {
  <img src="comments-placeholder.png">
}
```

Przykład z: https://blog.angular.io/introducing-angular-v17-4d7033312e4b

Deferrable views

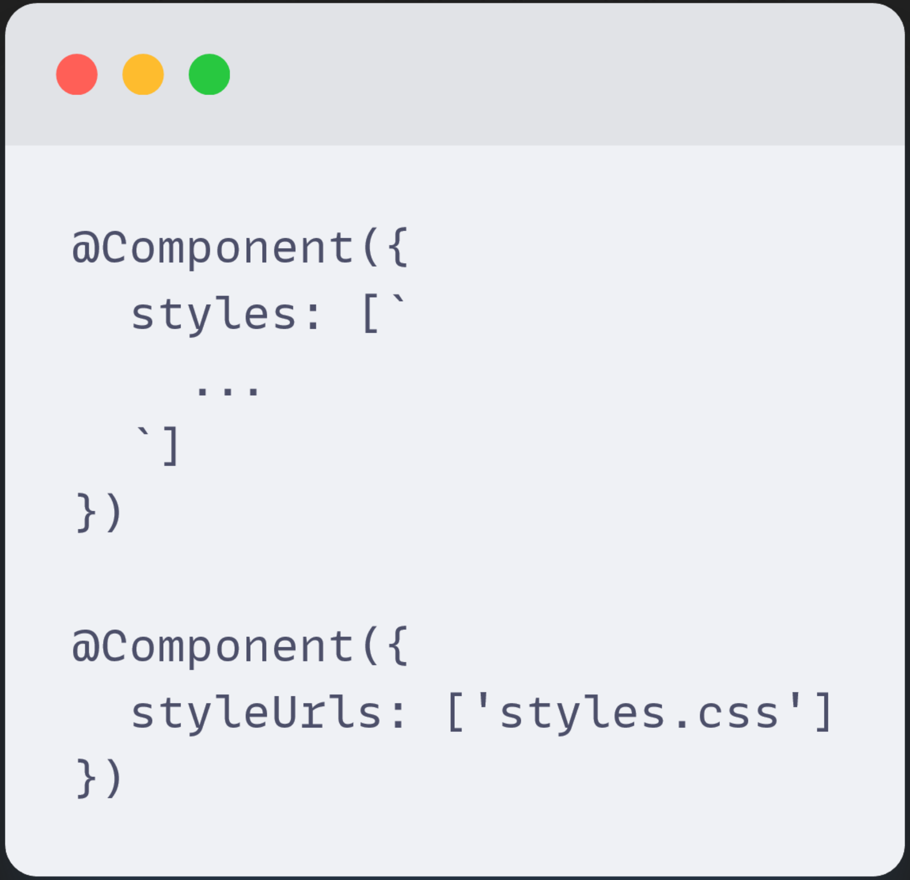
prefetch



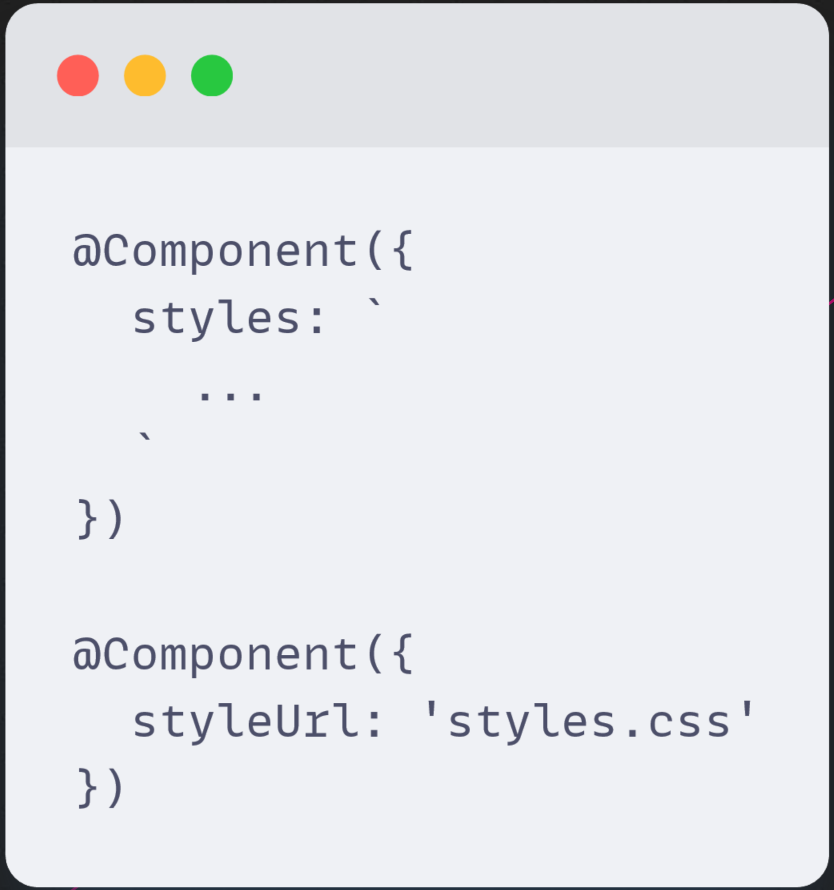
```
@defer (on interaction; prefetch on idle) {
  <calendar-cmp />
} @placeholder {
  <img src="placeholder.png" />
}
```

Materiał: https://angular.io/guide/defer

Styles and style



```
@Component({
  styles: [`
    ...
  `]
})

@Component({
  styleUrls: ['styles.css']
})
```



```
@Component({
  styles: `
    ...
  `
})

@Component({
  styleUrl: 'styles.css'
})
```

SSR and Hydration

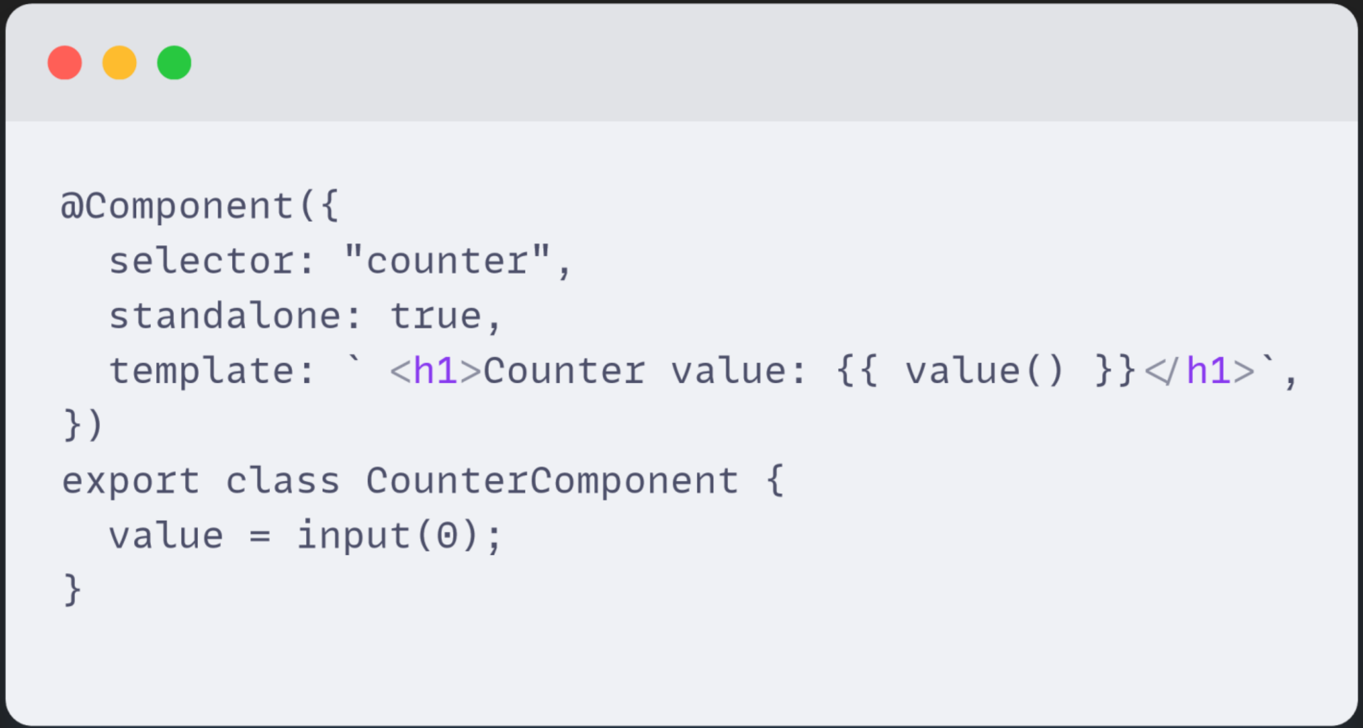
```
~ npm init @angular@next
```

Materiał: https://angular.io/guide/hydration

Signals

Signals

Signal-based inputs

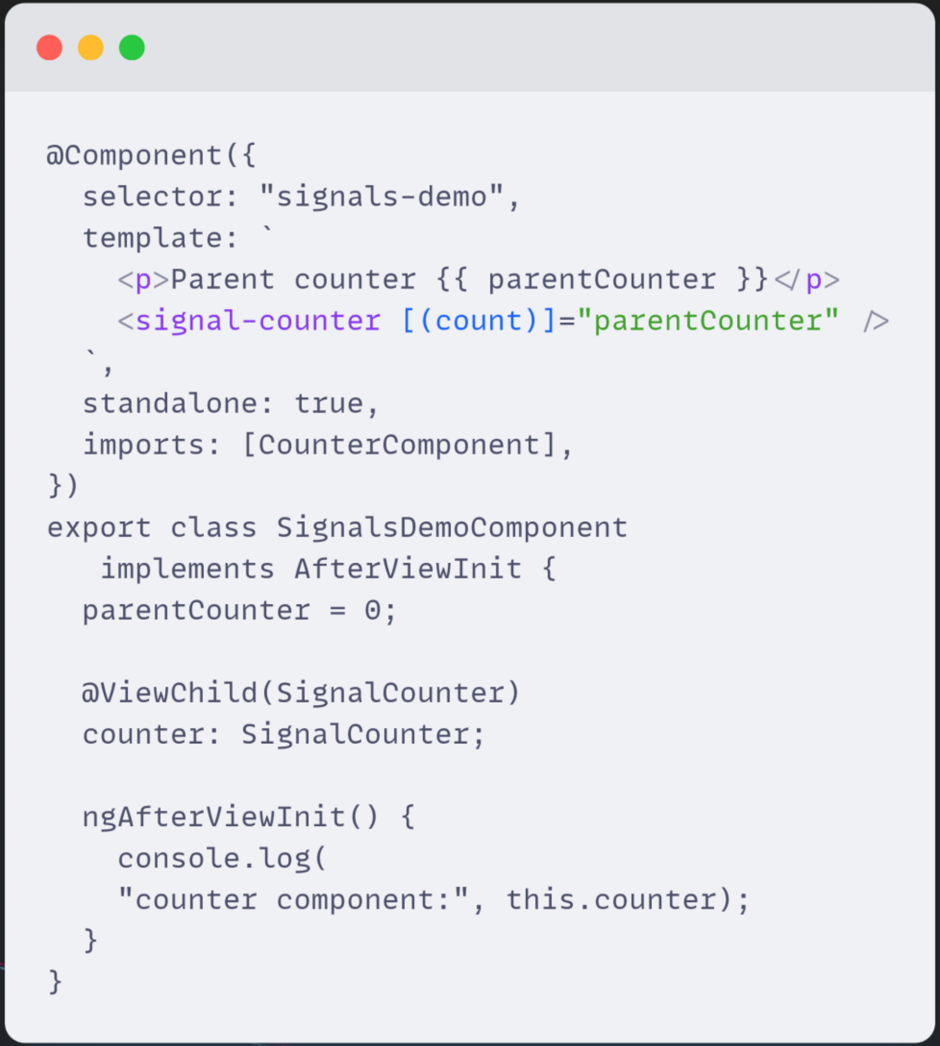


```
@Component({
  selector: "counter",
  standalone: true,
  template: ` <h1>Counter value: {{ value() }}</h1>`,
})
export class CounterComponent {
  value = input(0);
}
```

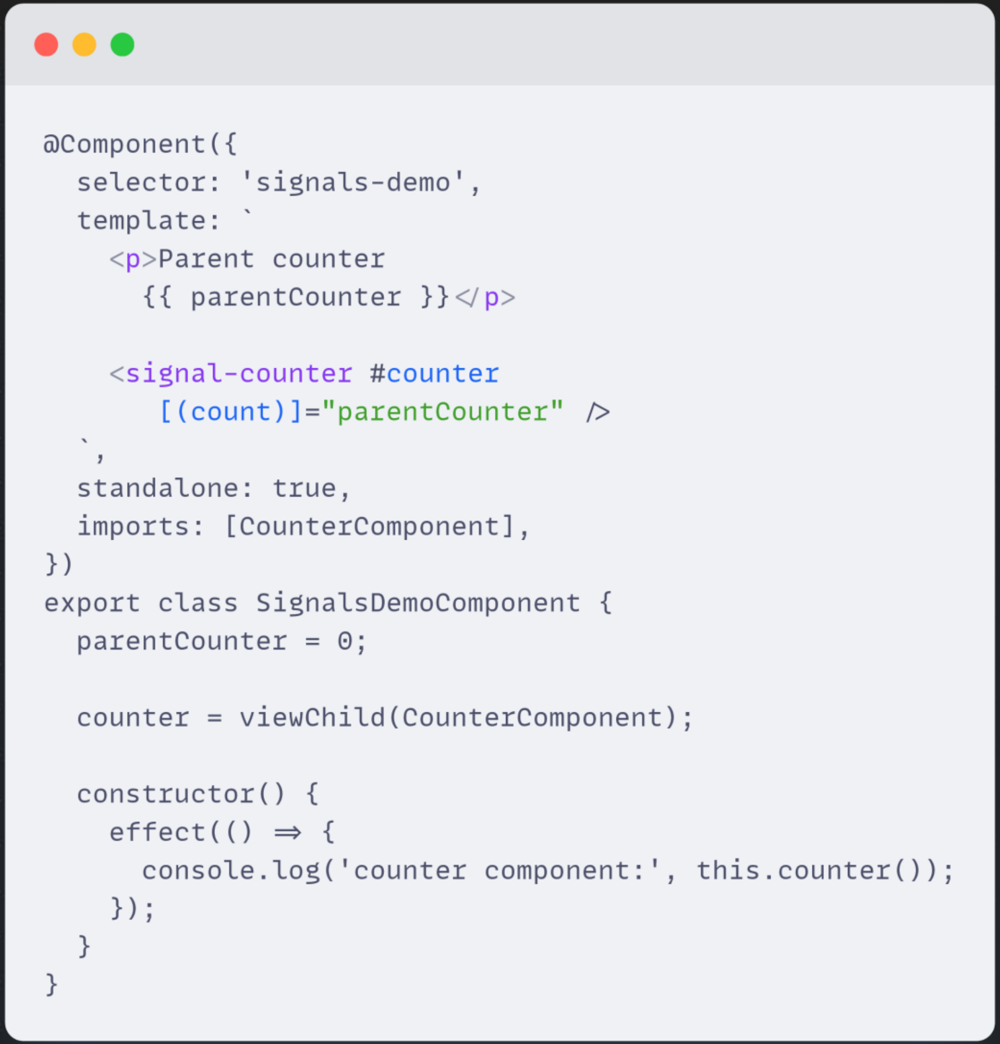
Przykład z: https://blog.angular-university.io/angular-signal-inputs

Signals

viewChild signal-base queries



```
@Component({
  selector: "signals-demo",
  template: `
    <p>Parent counter {{ parentCounter }}</p>
    <signal-counter [(count)]="parentCounter" />
  `,
  standalone: true,
  imports: [CounterComponent],
})
export class SignalsDemoComponent
   implements AfterViewInit {
  parentCounter = 0;

  @ViewChild(SignalCounter)
  counter: SignalCounter;

  ngAfterViewInit() {
    console.log(
    "counter component:", this.counter);
  }
}
```

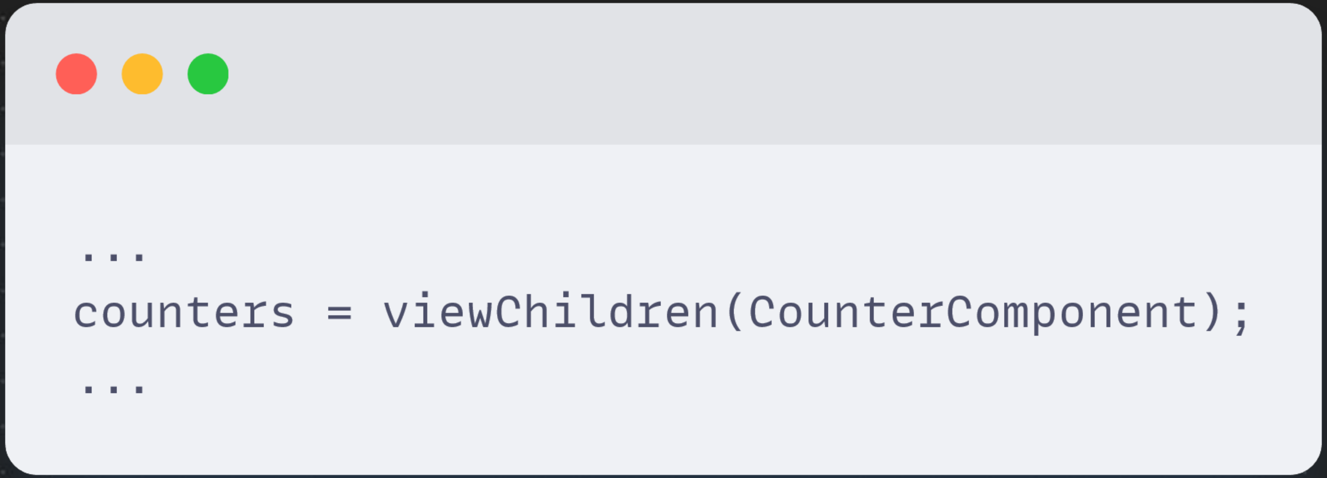


```
@Component({
  selector: 'signals-demo',
  template: `
    <p>Parent counter
      {{ parentCounter }}</p>

    <signal-counter #counter
       [(count)]="parentCounter" />
  `,
  standalone: true,
  imports: [CounterComponent],
})
export class SignalsDemoComponent {
  parentCounter = 0;

  counter = viewChild(CounterComponent);

  constructor() {
    effect(() => {
      console.log('counter component:', this.counter());
    });
  }
}
```

Przykład z: https://blog.angular-university.io/angular-signal-inputs

Signals

viewChildren, contentChild, contentChildren

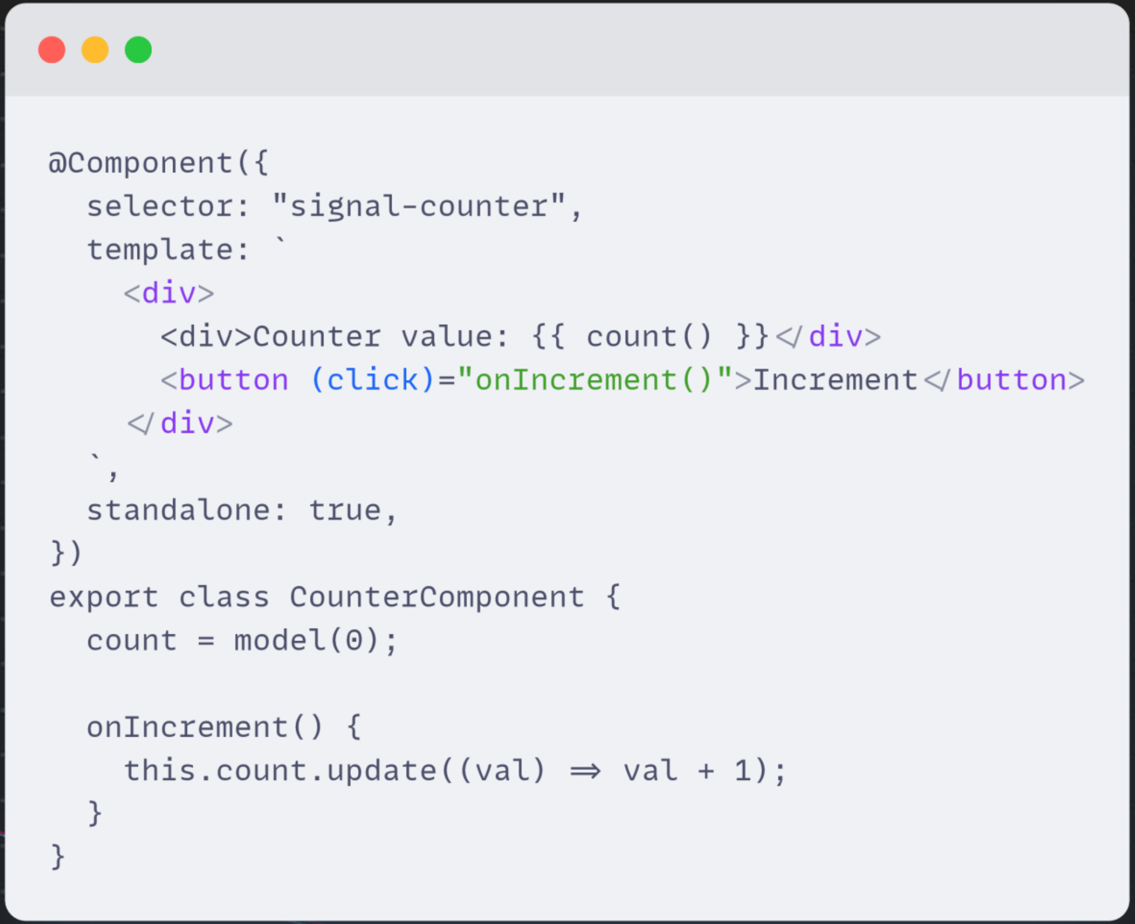


```
...
counters = viewChildren(CounterComponent);
...
```

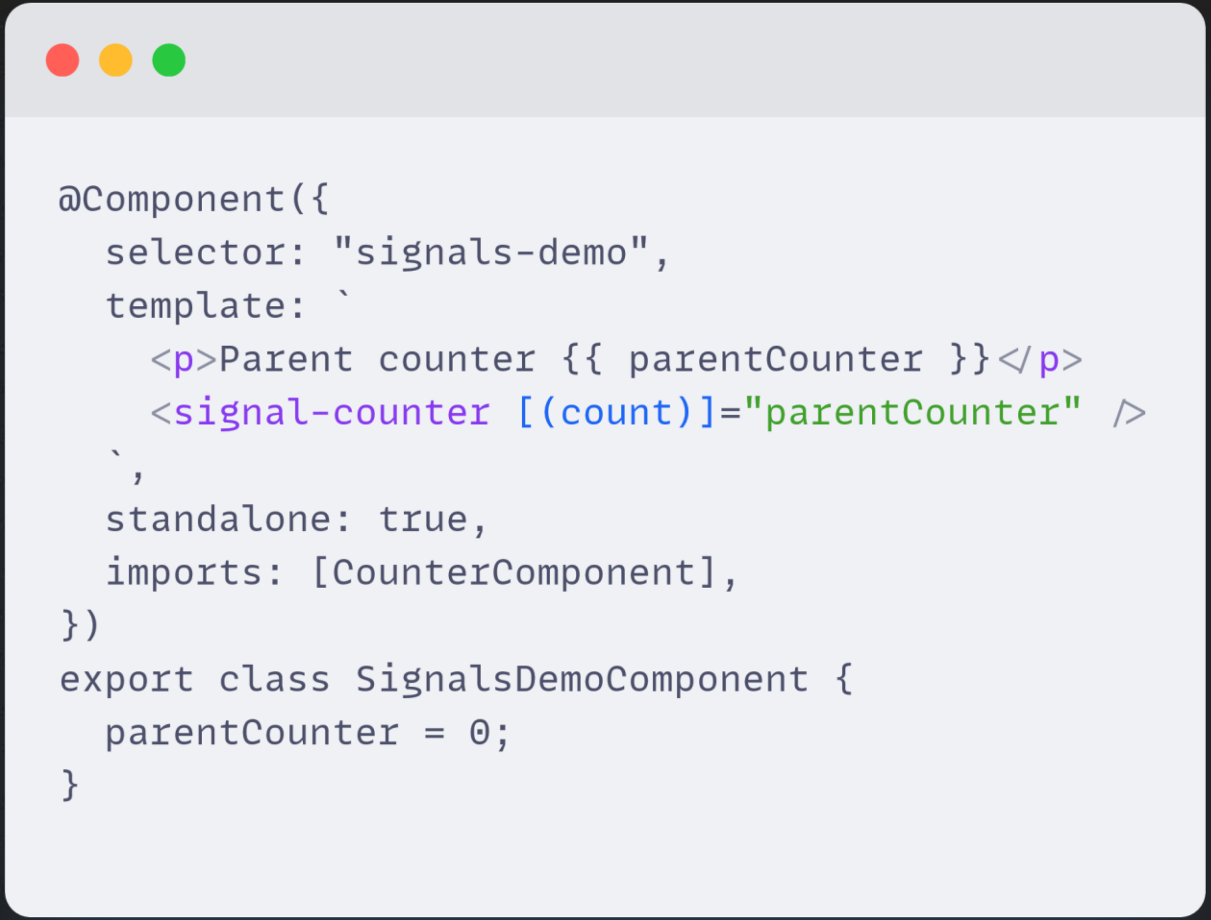
Przykład z: https://blog.angular-university.io/angular-signal-inputs

Signals

Model two-way binding



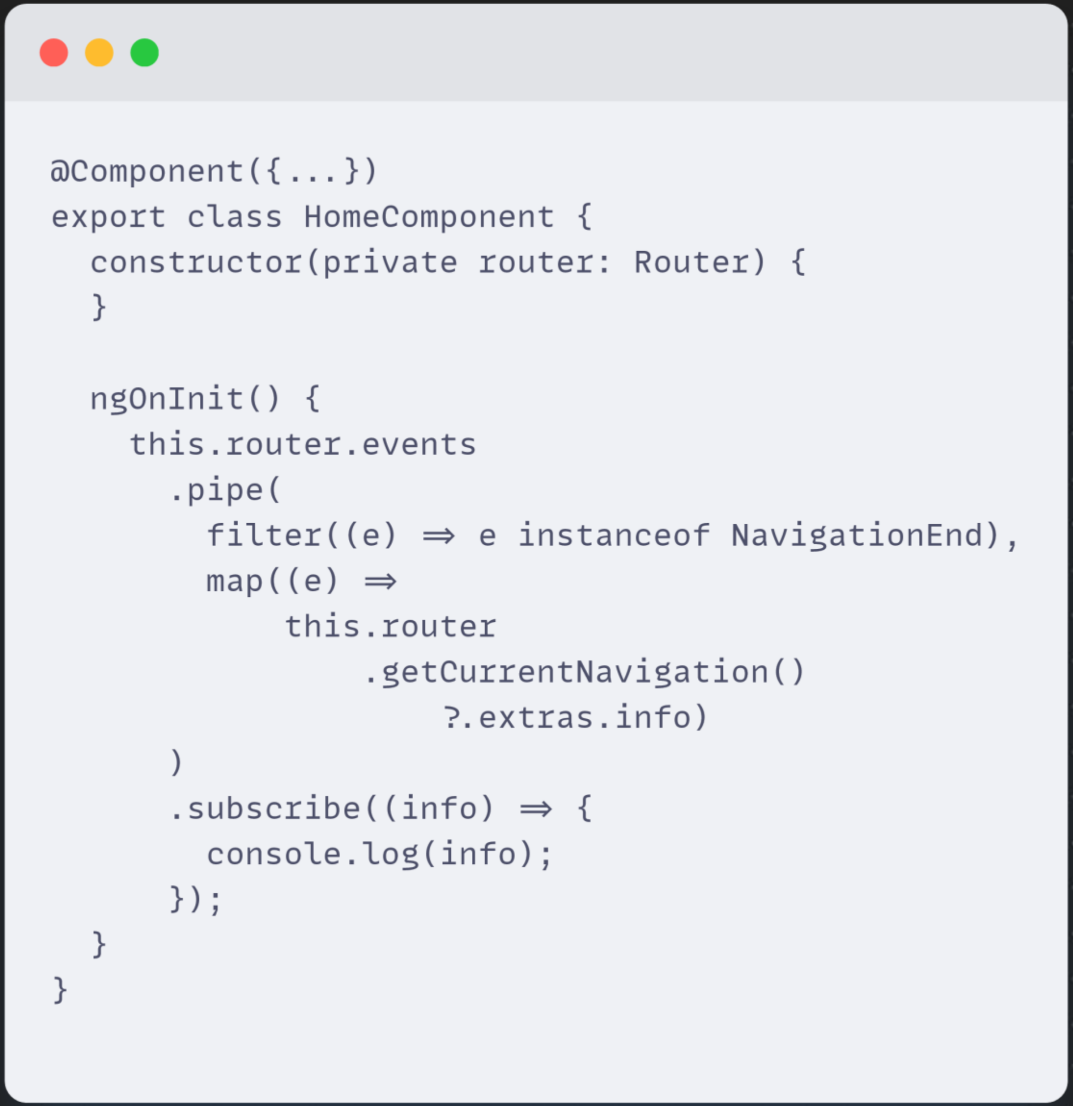
```
@Component({
  selector: "signal-counter",
  template: `
    <div>
      <div>Counter value: {{ count() }}</div>
      <button (click)="onIncrement()">Increment</button>
    </div>
  `,
  standalone: true,
})
export class CounterComponent {
  count = model(0);

  onIncrement() {
    this.count.update((val) => val + 1);
  }
}
```



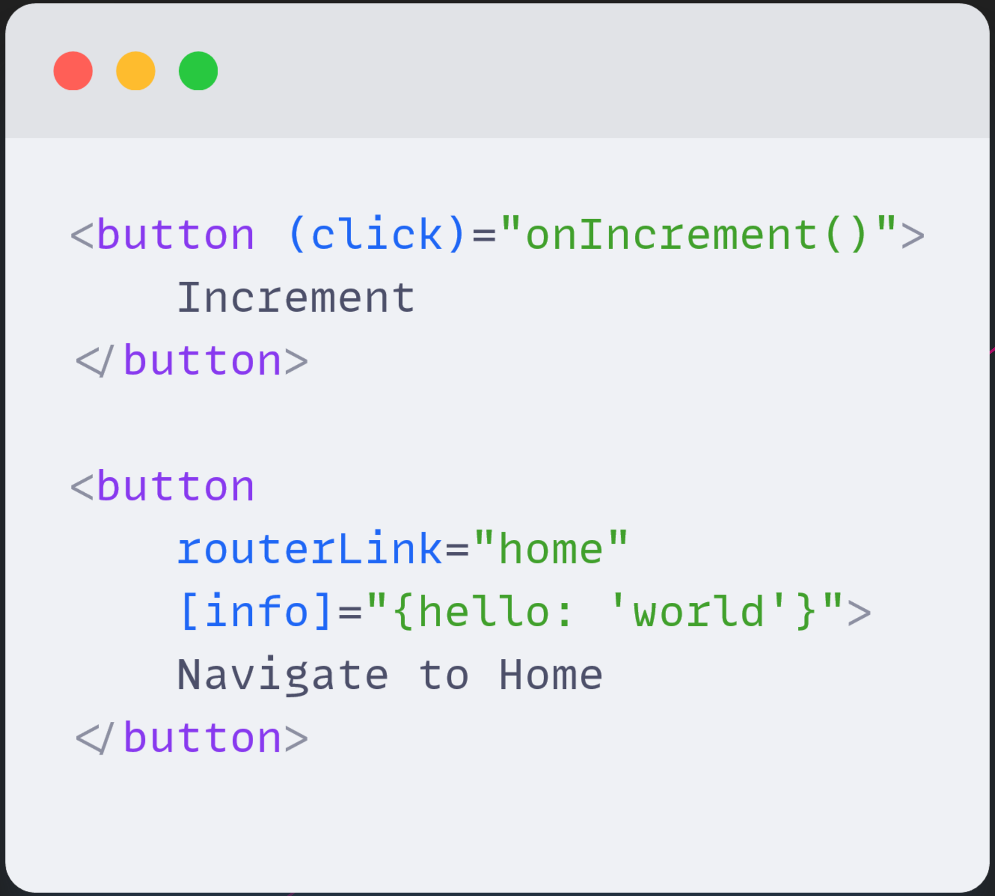
```
@Component({
  selector: "signals-demo",
  template: `
    <p>Parent counter {{ parentCounter }}</p>
    <signal-counter [(count)]="parentCounter" />
  `,
  standalone: true,
  imports: [CounterComponent],
})
export class SignalsDemoComponent {
  parentCounter = 0;
}
```

Przykład z: https://blog.angular-university.io/angular-signal-inputs

New router info parameter



```
@Component({...})
export class HomeComponent {
  constructor(private router: Router) {
  }

  ngOnInit() {
    this.router.events
      .pipe(
        filter((e) => e instanceof NavigationEnd),
        map((e) =>
            this.router
                .getCurrentNavigation()
                    ?.extras.info)
      )
      .subscribe((info) => {
        console.log(info);
      });
  }
}
```

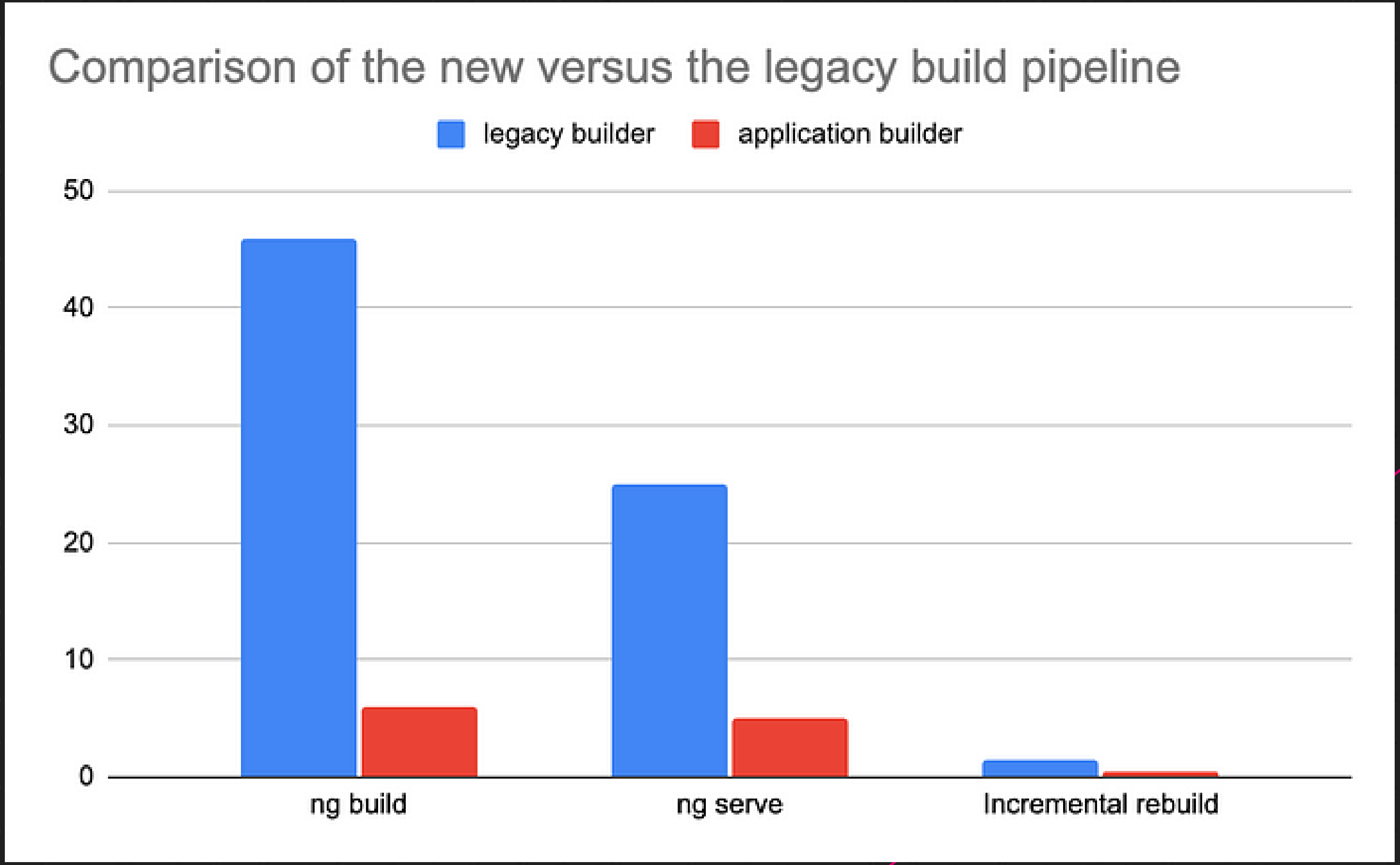


```
<button (click)="onIncrement()">
     Increment
</button>

<button
     routerLink="home"
     [info]="{hello: 'world'}">
     Navigate to Home
</button>
```

Przykład z: https://blog.angular.io/introducing-angular-v17-4d7033312e4b

Support for TypeScript 5.2

esbuild and vite



Przykład z: https://blog.angular.io/introducing-angular-v17-4d7033312e4b

czy warto?

synergy
codes

NEW COOL ANGULAR 17 FEATURES
ANGULAR DEV
NEW COOL ANGULAR 17 FEATURES
COMMERCIAL PROJECT STUCK ON ANGULAR 12
ANGULAR DEV
imgflip.com

To zależy

dzięki!

synergy
codes

pytania?

synergy
codes

Materiały

- https://blog.angular.io/introducing-angular-v17-4d7033312e4b
- https://www.youtube.com/watch?v=36Hcx7kRYDg
- https://blog.angular-university.io/angular-17-1-release/
- https://blog.angular-university.io/angular-17-2-release/
- https://krausest.github.io/js-framework-benchmark/current.html
- https://blog.angular-university.io/angular-signal-inputs
- https://angular.io/guide/hydration

synergy
codes

synergy
codes