



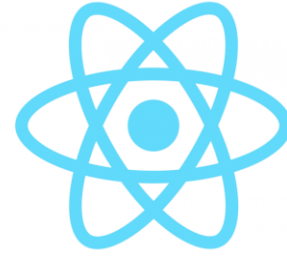
ANGULAR  
**ARCHITECTS**  
INSIDE KNOWLEDGE

# Web Components with Angular Elements

[angular-architects.io](https://angular-architects.io)

The JavaScript logo, consisting of the letters 'JS' in a bold, black, sans-serif font, centered within a bright yellow square. This square is positioned inside a white circle with a light gray border.

JS



Framework independent  
Components



@ManfredSteyer



All this can be  
polyfilled  
down to IE11  
(more or less)





# Angular Elements

# Install @angular/elements

```
ng add @angular/elements
```

This also installs polyfills for IE

# Angular Elements

```
@NgModule({  
  imports: [BrowserModule],  
  declarations: [DateComponent],  
  bootstrap: []  
})  
export class AppModule { }
```

# Angular Elements

```
@NgModule({  
  imports: [BrowserModule],  
  declarations: [DateComponent],  
  bootstrap: []  
})  
export class AppModule {  
  
  constructor(private injector: Injector) { }  
  
}
```



# Angular Elements

```
@NgModule({
  imports: [BrowserModule],
  declarations: [DateComponent],
  bootstrap: []
})
export class AppModule {

  constructor(private injector: Injector) {
  }

  ngDoBootstrap() {
    const DateElement = createCustomElement(
      DateComponent, { injector: this.injector });

    customElements.define('date-picker', DateElement);
  }
}
```



# Loading Web Components

```
<script src="bundle.js"></script>  
[...]
```

```
<date-picker [date]="myDate" (change)="handler()"></date-picker>
```

# Disable Checks for Web Components

```
@NgModule({  
  [...]  
  schemas: [CUSTOM_ELEMENTS_SCHEMA],  
})  
export class AppModule { }
```

# Dynamically Loading Web Components

```
const script = document.createElement('script');  
script.src = 'assets/bundle.js';  
document.body.appendChild(script);  
  
const clientA = document.createElement('date-picker');  
clientA['visible'] = true;  
document.body.appendChild(clientA);
```

# DEMO

# Compiling into one bundle

```
ng add ngx-build-plus --project project  
ng build project --prod --single-bundle
```



# Pros/Cons

- Combining different technologies und versions
- Overhead (Bundle Size)
- „Only“ Components
- Routing difficult