



```
"Hello Topicconf"
```



```
let greeting_var: string;
```

```
greeting_var = "Hello Topiconf";
```



```
let greeting_var: "Hello Topiconf";
```

```
greeting_var = "Hello Topiconf";
```



```
let greeting_var: "Hello Topiconf"  
                  | "Hello world";  
greeting_var = "Hello Topiconf";
```



```
let greeting_var: `Hello  
    ${"Topiconf"|"world"}`;  
greeting_var = "Hello Topiconf";
```

# who am i

- Sander Evers
- software engineer
- Topicus Education (2012)
- CumLaude team (2021)
- becoming a TypeScript fanboi

```
# js improvements over 10 years
```

```
[1,2,3].map(x => 10*x)
```

```
# js improvements over 10 years
```

```
[1,2,3].map(x => 10*x)
```

```
queryParams = {...defaultParams, max:100}
```



# js improvements over 10 years

```
[1,2,3].map(x => 10*x)
```

```
queryParams = {...defaultParams, max:100}
```

```
data = fetch().then(resp => parse(resp))
```

# js improvements over 10 years

```
[1,2,3].map(x => 10*x)
queryParams = {...defaultParams, max:100}
data = fetch().then(resp => parse(resp))
while (true)
  item = await queue.read()
```

## # js improvements over 10 years

```
[1,2,3].map(x => 10*x)
queryParams = {...defaultParams, max:100}
data = fetch().then(resp => parse(resp))
while (true)
  item = await queue.read()
while (true)
  yield i++
```

# js improvements over 10 years

```
[1,2,3].map(x => 10*x)
queryParams = {...defaultParams, max:100}
data = fetch().then(resp => parse(resp))
while (true)
  item = await queue.read()
while (true)
  yield i++
item?.title?.toUpperCase() ?? 'Untitled'
```

# js improvements over 10 years

```
[1,2,3].map(x => 10*x)
queryParams = {...defaultParams, max:100}
data = fetch().then(resp => parse(resp))
while (true)
  item = await queue.read()
while (true)
  yield i++
item?.title?.toUpperCase() ?? 'Untitled'
`${n} bottles of ${item} on the wall`
```

```
# js improvements over 10 years
```

```
[1,2,3].map(x => 10*x)  
queryParams = {...defaultParams, max:100}  
data = fetch().then(resp => parse(resp))  
while (true)  
  item = await queue.read()  
while (true)  
  yield i++  
item?.title?.toUpperCase() ?? 'Untitled'  
`${n} bottles of ${item} on the wall`  
import / export
```

# js improvements over 10 years

```
[1,2,3].map(x => 10*x)
queryParams = {...defaultParams, max:100}
data = fetch().then(resp => parse(resp))
while (true)
  item = await queue.read()
while (true)
  yield i++
item?.title?.toUpperCase() ?? 'Untitled'
`${n} bottles of ${item} on the wall`
import / export
class TodoList extends Component {}
```

```
# but still a crappy standard library
```

```
[1,5,10].sort()
```

```
_.sortBy([1,5,10])
```



TypeScript = JavaScript + type annotations

```
type Fraction = {num: number, den: number}
```

```
function multiply(a: Fraction, b: Fraction): Fraction {  
  return {  
    num: a.num * b.num,  
    den: a.den * b.den  
  };  
}
```

## # why typing

- QA: prove absence of errors
- documentation
- maintenance / refactoring
- code completion

```
type SimpleNounPhrase = `${Particle} ${Noun}`;
```

```
type Particle = 'de' | 'het' | 'een';
```

```
type Noun = FM | N;
```

```
type FM = 'boom' | 'roos' | 'vis';
```

```
type N = 'huis' | 'raam';
```

```
type SimpleNounPhrase = `${Particle} ${Noun}`;
```

```
type Particle = 'de' | 'het' | 'een';
```

```
type Noun = FM | N;
```

```
type FM = 'boom' | 'roos' | 'vis';
```

```
type N = 'huis' | 'raam';
```

```
const ex1: SimpleNounPhrase = 'raam het'; ❌
```

```
type SimpleNounPhrase = `${Particle} ${Noun}`;
```

```
type Particle = 'de' | 'het' | 'een';
```

```
type Noun = FM | N;
```

```
type FM = 'boom' | 'roos' | 'vis';
```

```
type N = 'huis' | 'raam';
```

```
const ex1: SimpleNounPhrase = 'raam het'; ❌
```

```
const ex2: SimpleNounPhrase = 'het raam'; ✅
```

```
type SimpleNounPhrase = `${Particle} ${Noun}`;
```

```
type Particle = 'de' | 'het' | 'een';
```

```
type Noun = FM | N;
```

```
type FM = 'boom' | 'roos' | 'vis';
```

```
type N = 'huis' | 'raam';
```

```
const ex1: SimpleNounPhrase = 'raam het'; ❌
```

```
const ex2: SimpleNounPhrase = 'het raam'; ✅
```

```
const ex3: SimpleNounPhrase = 'de raam'; ✅
```

```
type ParticleMapping = { de: FM; het: N; een: FM | N };
```

```
type ParticleMapping = { de: FM; het: N; een: FM | N };
```

```
const ex4: ParticleMapping = {  
  de: 'boom',  
  het: 'huis',  
  een: 'vis'  
}
```



```
type ParticleMapping = { de: FM; het: N; een: FM | N };
```

```
const ex4: ParticleMapping = {  
  de: 'boom',  
  het: 'huis',  
  een: 'vis'  
}
```

```
type CorrectMap = { [p in Particle]: `${p} ${ParticleMapping[p]}` };
```

```
type ParticleMapping = { de: FM; het: N; een: FM | N };
```

```
const ex4: ParticleMapping = {  
  de: 'boom',  
  het: 'huis',  
  een: 'vis'  
}
```

```
type CorrectMap = { [p in Particle]: `${p} ${ParticleMapping[p]}` };
```

```
==> { de: `de ${FM}`; het: `het ${N}`; een: `een ${FM | N}` }
```

```
type ParticleMapping = { de: FM; het: N; een: FM | N };
```

```
const ex4: ParticleMapping = {  
  de: 'boom',  
  het: 'huis',  
  een: 'vis'  
}
```

```
type CorrectMap = { [p in Particle]: `${p} ${ParticleMapping[p]}` };
```

```
==> { de: `de ${FM}`; het: `het ${N}`; een: `een ${FM | N}` }
```

```
const ex5: CorrectMap = {  
  de: 'de roos',  
  het: 'het raam',  
  een: 'een huis'  
}
```

```
type CorrectNounPhrase = CorrectMap[keyof CorrectMap];
```

```
type CorrectNounPhrase = CorrectMap[keyof CorrectMap];
```

```
const ex6: CorrectNounPhrase = 'de raam';  
const ex7: CorrectNounPhrase = 'het raam';
```



```
type CorrectNounPhrase = CorrectMap[keyof CorrectMap];
```

```
const ex6: CorrectNounPhrase = 'de raam';  
const ex7: CorrectNounPhrase = 'het raam';
```



```
const ex7: CorrectNounPhrase = 'h';
```

- het huis
- het raam
- een huis

Press ^. to choose the selected (or

Doorstroom

Instroom

Uitstroom

Geslaagd

Overgang

Filters



Voorgestelde filters

Alle filters

Schooljaar

2019/2020

Vestiging

Niveau

VMBO b

VMBO k

VMBO (gt)

VMBO (gt) / HAVO

HAVO

HAVO / VWO

VWO

OND

VMBO b/k

Leerjaar

Doorstroom

Doorstroom

Filters

Filters resetten

2019/2020

Groepering

Niveau



Leerjaar



(kies...)



Toon B

Niveau ↑	Leerjaar		Percentage
VMBO b	1	<div><div>b2</div><div>k2</div><div>b1</div></div>	92%
	2	<div><div>b3</div><div>k3</div></div>	96%
	3	<div><div>b4</div><div>b3</div></div>	88%
	4	<div><div>Geslaagd</div></div>	96%
VMBO k	1	<div><div>b2</div><div>k2</div><div>t2</div></div>	98%
	2	<div><div>b3</div><div>k3</div><div>k2</div></div>	92%
	3	<div><div>b4</div><div>k4</div></div>	94%
	4	<div><div>Geslaagd</div></div>	98%
VMBO (gt)	1	<div><div>k2</div><div>t2</div></div>	98%
	2	<div><div>k3</div><div>t3</div></div>	96%
	3	<div><div>t4</div></div>	94%

```
interface FacLbLoopbaan {  
    lb_d_inschrijving?: string;  
    lb_d_uitschrijving?: string;  
    lb_fk_br_vest?: DimBrBrin;  
    lb_fk_br_vest_volgend_pl?: DimBrBrin;  
    lb_fk_ilt?: DimIltOpleidingscode;  
    lb_fk_ilt_volgend_pl?: DimIltOpleidingscode;  
    ...  
}
```

```
interface DimIltOpleidingscode {  
    ilt_abb_profiel?: string;  
    ilt_co_niveau?: number;  
    ilt_nm_niveau?: Niveau;  
    ...  
}
```



```
interface FacLbLoopbaan {  
    lb_d_inschrijving?: string;  
    lb_d_uitschrijving?: string;  
    lb_fk_br_vest?: DimBrBrin;  
    lb_fk_br_vest_volgend_pl?: DimBrBrin;  
    lb_fk_ilt?: DimIltOpleidingscode;  
    lb_fk_ilt_volgend_pl?: DimIltOpleidingscode;  
    ...  
}
```

```
interface DimIltOpleidingscode {  
    ilt_abb_profiel?: string;  
    ilt_co_niveau?: number;  
    ilt_nm_niveau?: Niveau;  
    ...  
}
```

```
doorstroomFilters: FilterName[] = [  
    'x_schooljaar_historie',  
    'x_multiselect_schooljaar',  
    'lb_nm_vestiging',  
    'lb_fk_ilt.ilt_nm_niveau',  
    'lb_nm_leerjaar',  
    'lb_nm_idu',  
];
```

...

```
<app-toggle-buttons class="right"
  [value]="variant"
  (valueChange)="qp.dispatch('variant',$event)">
</app-toggle-buttons>
```

Actueel

Historie

doorstroom.component.html

...

```
<app-toggle-buttons class="right"
  [value]="variant"
  (valueChange)="qp.dispatch('variant',$event)">
</app-toggle-buttons>
```

enum DashboardVariant

Actueel

Historie

doorstroom.component.html

...

```
<app-toggle-buttons class="right"
  [value]="variant"
  (valueChange)="qp.dispatch('variant',$event)">
</app-toggle-buttons>
```

enum DashboardVariant

Actueel

Historie

query-param.service.ts

```
const configs = {
  adviesType: attrPathCodec,
  'brin-overgang': booleanCodec,
  col: multiCodec<CijferColumn>(),
  threshold: numberCodec,
  variant: singleCodec<DashboardVariant>(),
};
```

```
dispatch<T extends keyof typeof configs>
  (paramName: T, value: Parameters<typeof configs[T]['encode']>[0])
{
  ...
}
```



"Thank you Topiconf"