

Variabler och konstanter

Variabler

- ✓ Program kommer ihåg värden med hjälp av variabler.
- ✓ Genom att binda ett namn till ett värde (tilldela variabeln ett värde) kan du referera till värdet i kod du skriver.
- ✓ En variabel deklarerar med hjälp av nyckelordet `let` (innan ES6 användes `var`) följt av namnet på variabeln.

```
let count                // undefined  
let result = (4 + 8) * 3 + 8 // 44
```

- ✓ Efter att en variabel deklarerar kan den användas i uttryck.

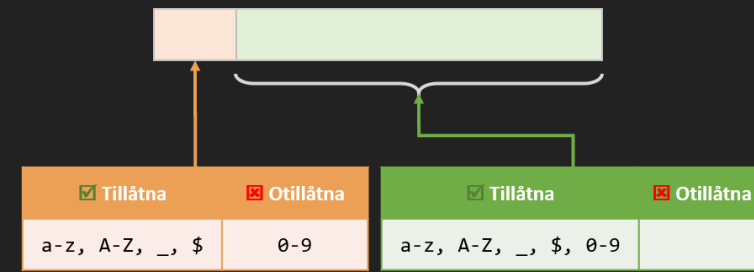
```
let result = (4 + 8) * 3 + 8 // 44  
let theMeaningOfLife = result - 2 // 42
```

- ✓ En variabels värde kan ändras.

```
let result = (4 + 8) * 3 + 8 // 44  
result = 13                // 13  
result = result + 29       // 42
```

Variablers namn

- ✓ Beskrivande namn är centralt. Håll dig till ett språk, gärna engelska.
- ✓ Kan vara vilket ord som helst bara det inte är ett nyckelord eller ett reserverat ord.
- ✓ Får innehålla tecknen a-z, A-Z, 0-9, _ och \$.
 - Får inte börja med en siffra.
 - Fler tecken är tillåtna, som t.ex. åäö, men använd aldrig dessa.
- ✓ Kan bestå av flera ord. Tillämpa i så fall så kallad kamelnotation ("*camelCasing*").
 - Ett sätt att skriva samman ord utan bindestreck eller mellanslag.
 - Varje ord, utom det första, inleds med versal.



```
let bestBookEver = 'The Hitchhiker\'s Guide to the Galaxy'
```

```
let theMeaningOfLife = 42
```

Nyckelord och reserverade ord

- ✓ Problem vid definition av en variabel? Du kanske har använt något av orden nedan.

<code>await</code>	<code>delete</code>	<code>import</code>	<code>this</code>
<code>break</code>	<code>do</code>	<code>in</code>	<code>throw</code>
<code>case</code>	<code>else</code>	<code>instanceof</code>	<code>typeof</code>
<code>catch</code>	<code>export</code>	<code>new</code>	<code>try</code>
<code>class</code>	<code>extends</code>	<code>let</code>	<code>var</code>
<code>const</code>	<code>finally</code>	<code>return</code>	<code>void</code>
<code>continue</code>	<code>for</code>	<code>static</code>	<code>while</code>
<code>debugger</code>	<code>function</code>	<code>super</code>	<code>with</code>
<code>default</code>	<code>if</code>	<code>switch</code>	<code>yield</code>
<code>enum</code>	<code>interface</code>	<code>private</code>	<code>public</code>
<code>implements</code>	<code>package</code>	<code>protected</code>	

Konstanter

- ✓ Konstanter fungerar precis som variabler med skillnaderna:
 - En konstant måste initieras till värde då den deklarerar.
 - En konstants värde kan inte ändras.
- ✓ En konstant deklarerar med hjälp av nyckelordet `const` följt av namnet på konstanten och dess värde.

```
const myFavoriteNumber = 42  
const MY_FAVORITE_NUMBER = 42
```

Det finns olika uppfattningar om hur konstanter ska skrivas!

- ✓ Det blir fel, kastas ett undantag, om du försöker ändra på en konstants värde.

```
const MY_FAVORITE_NUMBER = 42  
MY_FAVORITE_NUMBER = 21  
// MY_FAVORITE_NUMBER = 21  
//           ^  
//  
// TypeError: Assignment to constant variable.
```

Variabler och konstanter räckvidd (”scope”)

- ✓ En variablers räckvidd (”scope”) är det område där variabeln kan användas.
- ✓ Variablers, och konstanter, räckvidd är ”block scope”, vilket innebär att de bara kan användas inom det område, block, där de har deklarerats.
 - Ett block med kod är, lite förenklat, kod som står inom ett par klammerparenteser.

```
{  
  let bestBookEver = 'The Hitchhiker\'s Guide to the Galaxy'  
  console.log(bestBookEver)  
}  
console.log(bestBookEver) // Utanför blocket kan inte variabeln användas och ett fel inträffar.  
  
// OUTPUT:  
// The Hitchhiker's Guide to the Galaxy  
// Uncaught ReferenceError: bestBookEver is not defined
```

