

JavaScript alkeet

Lausekkeet
Operaattorit
Tietotyypin muunnos
Ehtolauseet
Funktiot

Metropolia 2011

Lausekkeet

- operaattorit $+, -, =, >, <$
- operandit
- esim. var summa = $4 + 3$;
- muuttuja, operaattori, operandi

Operaattorit

- binääriset operaattorit: kaksi operandia
- unaariset operaattorit: yksi operandi
- sijoitus-, vertailu-, bitti-, erikois- ja merkkijono- operaattorit
- aritmeettiset ja loogiset operaattorit
- merkkijonoja yhdistävä operaattori +

Suoritusjärjestys

- () sulkeet (vas->oik)
- ++ -- auto increment (lisäys, vähennys merkit) (oik-> vas)
- ! Looginen EI (oik-> vas)
- */ % kertolasku, jakolasku, modulo (jakojäännös) (vas oik)
- +-, plus ja miinus

Suoritusjärjestys 2.

- + concatenation (merkkijonojen yhdistäminen)
- < <=
- > >=
- == !=
- === !== identtisiä (sama tyyppi)
- && logical AND

Operaattoreita

- laskutoimitukset $+$, $-$, $*$, $/$, $\%$
- unaariset: $+=$, $-=$, $*=$, $/=$
- edessä oleva kasvatusoperaattori $++x$
- jäljessä oleva kasvatusoperaattori $x++$

tietotyypin muunnos

- `var mj="merkkijono", luku = 100;`
- `var t1 = mj + luku`
- `t1 = "merkkijono100"`
- `"22"+4` `-> 224`
- `"22"*1+4` `-> 26` `!!!!`
- `parseInt()`
- `parseFloat()`

vertailu operaattorit

- `< > ==`
- operandeja voivat olla numerot tai merkkijonot
- tulos on totta tai epätotta (boolean arvo)
- esim. `"tom" < "tommi" -> totta`
- js konsoli test
- `===` sama arvo ja sama tietotyyppi

Loogiset operaattori

- Looginen AND
 - $a \ \&\& \ b$ (rahaa $\&\&$ aikaa)
- Looginen OR
 - $a \ || \ b$ (esim. $ikä < 0 \ || \ > 130$)
- Looginen EI
 - $!a$

Ehdollinen operaattori:

- $(\text{ehto}) ? a : b;$
- $\text{Bigger} = (x > y) ? x : y;$

Tietotyypin muunnos metodeja

- `parseInt()` muuttaa merkkijonon numeroksi
- `parseFloat()` muuttaa merkkijonon desimaaliluvuksi

eval() metodi

- eval() metodi suorittaa merkkijonon kuten se olisi ohjelmakoodia (merkkijono ”evaluoidaan” lausekkeena)

EHTOLAUSEET

if lause:

```
If (ehto) {  
    Lauseet;  
}
```

```
if ( ika > 17) {  
    alert("Olet täysi-ikäinen")  
}
```

IF ELSE

```
if (ehto) {  
    ohjelmalauseet;  
}  
else  
{  
    ohjelmalauseet;  
}
```

ELSE IF lause

```
if (ehto) {  
    lauseet1;  
}  
else if (ehto) {  
    lauseet2;  
}
```

Switch lause

```
switch (lauseke) {  
    case label:  
        lauseet;  
        break;  
    case label2:  
        lauseet;  
        break;  
default: lause;  
}
```

Funktiot

- Funktiot ovat itsenäisiä ohjelmakokonaisuuksia, jotka suorittavat jonkin tietyn tehtävän
- Niitä voidaan käyttää yhä uudelleen, jolloin voidaan välttää samankaltaisten ohjelmarakenteiden toistoa
- Niitä käytetään myös ohjelman osittamiseen pienemmiksi moduleiksi

Funktio

- Funktio on komentoryhmä, joka tekee tietyn tehtävän ja palauttaa arvon.
- Funktion ja metodin ero: metodeja käytetään olioiden yhteydessä
- Ennen kuin funktiota voidaan kutsua Funktio pitää määritellä

Funktion määrittely

```
function funktion_nimi() { lause;  
lause;}
```

```
function funktion_nimi(parametri,  
parametri2) { lause; lause;}
```

es. 7.1.

Funktio

- Funktio voidaan kutsua suoraan linkistä käyttäen JavaScript pseudoprotokollaa (es. 7.3)
- Funktion kutsuminen tapahtumasta (tapahtuman käsittelijä) (es. 7.4)

Funktiot

- globaalit ja paikallinen muuttujat (es. 7.5.)
- funktio voi palauttaa arvon (es. 7.6)
- **rekursio:**
- rekursion määritelmä:
- rekursio: katso rekursion määritelmä