

PHP 01

Operadors lògics i de comparació

Operadors aritmètics

Logical Operators		
Example	Name	Result
<code>\$a and \$b</code>	And	TRUE if both <code>\$a</code> and <code>\$b</code> are TRUE .
<code>\$a or \$b</code>	Or	TRUE if either <code>\$a</code> or <code>\$b</code> is TRUE .
<code>\$a xor \$b</code>	Xor	TRUE if either <code>\$a</code> or <code>\$b</code> is TRUE , but not both.
<code>! \$a</code>	Not	TRUE if <code>\$a</code> is not TRUE .
<code>\$a && \$b</code>	And	TRUE if both <code>\$a</code> and <code>\$b</code> are TRUE .
<code>\$a \$b</code>	Or	TRUE if either <code>\$a</code> or <code>\$b</code> is TRUE .

4) Operadors lògics i de comparació

5) Operadors aritmètics

- Operadors de **comparació**:
 - Permeten comparar dos valors del **mateix tipus** (o **assimilable**).
 - El resultat és un valor **booleà** (**true**, **false**).
 - Servirà per prendre **decisions**.
 - Per exemple per utilitzar-les en una estructura condicional **if .. else**.
- Veiem la taula següent:

Nom	Exemple	Resultat
Més gran que	$\$a > \b	true si \$a és major que \$b false en cas contrari
Més xicotet que	$\$a < \b	true si \$a és menor que \$b false en cas contrari
Major o igual que	$\$a \geq \b	true si \$a és major o igual que \$b false en cas contrari
Menor o igual que	$\$a \leq \b	true si \$a és menor o igual que \$b false en cas contrari

Nom	Exemple	Resultat
Diferent	$\$a <> \b o $\$a \neq \b	true si \$a és diferent a \$b false en cas contrari
Idèntic o estrictament igual	$\$a === \b	true si \$a és igual a \$b i sn del mateix tipus false en cas contrari
No idèntic o estrictament diferent	$\$a !== \b	true si \$a no és igual a \$b o no són del mateix tipus false en cas contrari
Igual	$\$a == \b	true si \$a és igual a \$b false en cas contrari

- **Exemple** donada la variable `$text1 = "1"`. **Operadors especials**:
 - `===` "és estrictament igual" i `!==` "no és estrictament igual".
 - `$text1 === 1`, retornarà **false**:
 - No és igual perquè un **text no és igual a un nombre**.
 - `$text1 == 1`, retornarà **true**:
 - La **comparació no és estricta**, tracta de **realitzar automàticament conversions** per comprovar si es pot establir una **equivalència entre els dos valors**.
 - Es **busca l'equivalent numèric** del text:
 - Després es fa la **comparació**: s'obté **true**.

- **Exemple Operadors especials:**
 - Si es compara un **número** amb una **cadena de caràcters** o la comparació **implica cadenes de caràcters numèriques**:
 - Cada cadena de caràcters és convertida en un nombre.
 - La **comparació** es realitzarà **numèricament** :
 - Excepte quan l'operador utilitzat és **===** o **!==**:
 - En aquests casos, també **comparem el tipus**.

- Operadors **lògics**:
 - Permeten crear **condicions** per a les diferents estructures en PHP.
 - En estructures:
 - **Condicionals**. if, if .. else, if.. else if .. else
 - **Repetitives**. while, do.. while, for
 - Els operadors lògics més importants són **and** (&&) i **or** (||).
 - Els operands són valors booleans (**true**, **false**).
 - Veiem la taula següent:

- Taula de veritat:

Operands		Resultats de l'operació			
A	B	A && B	A B	! A	A xor B
false	false	false	false	true	false
true	false	false	true	false	true
false	true	false	true	true	true
true	true	true	true	false	false

Nom	exemple	Resultat
and o &&	<code>\$a and \$b</code> <code>\$a && \$b</code>	true si \$a és true i \$b és true false en cas contrari
or o	<code>\$a or \$b</code> <code>\$a \$b</code>	true si \$a o \$b és true , o tots dos false en cas contrari
xor	<code>\$a xor \$b</code>	true si \$a o \$b és true , per no ambdós false en cas contrari
not o !	<code>!\$a</code> <code>not \$a</code>	true si \$a no és true false en cas contrari

- Una condició pot ser **tan llarga com es vulga o necessite** i usar **tants operadors lògics com es vulga o necessiten**.
- L'ús dels **parèntesis** atorgarà la **prioritat de l'execució d'unes operacions i altres** (com en matemàtiques).

- **Expressions** amb **operadors lògics i relacionals**:
 - Tornen com a resultat valor **booleà** (**true**, **false**).
- Per exemple:
 - Si **\$a = true** i **\$b = false** l'expressió **\$a && \$b** retorna **false**:
 - És falsa perquè no es compleix que \$a i \$b siguin veritat.
 - Si **\$a = true** i **\$b = false** l'expressió **\$a || \$b** retorna **true**:
 - Almenys un dels dos operands és veritable.
 - Si **\$a = true** l'expressió **!\$a** retorna **false** (l'oposat o contrari).

| : ALT GR + 1

- **&&** i **||** : **operadors en curtcircuit**.

- Si **no es compleix** la condició d'un terme:

- **No s'avalua la resta** de l'operació.

- Per **exemple**:

`($a == $b && $c != $d && $h >= $k)`, té tres avaluacions:

- La primera comprova si la variable a és igual a b.

- Si **no es compleix** aquesta condició:

- El **resultat** de l'expressió és **false**:

- **No s'avaluen les altres dues** condicions posteriors.

- **&&** i **||** : **operadors en curtcircuit**:

- Per **exemple**:

`($a < $b || $c != $d || $h <= $k)`

- Primer s'avalua si **\$a** és menor que **\$b**:
 - Si **es compleix** aquesta condició:
 - El **resultat** de l'expressió és **true**:
 - **No s'avaluen les altres dues** condicions posteriors.

Operador de negació aplicat sobre nombres o text

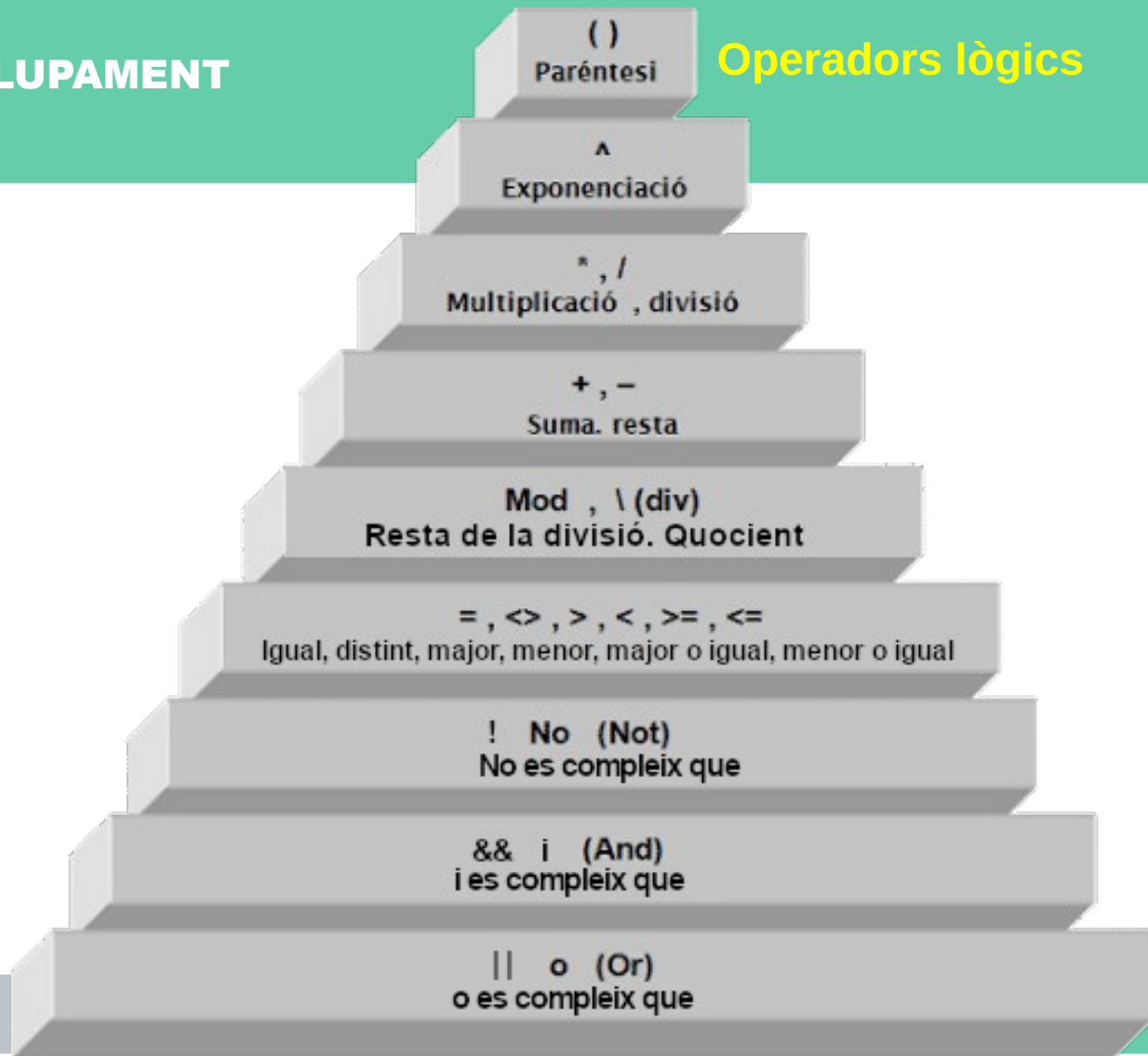
- Si `$a = true`, aleshores `!$a` retorna `false`.
- Què passa si `$a` és un `nombre` o un `text` en aquesta expressió?
- Si `$a` és un `nombre`:
 - Equival a `false` si el seu **valor numèric és 0**.
 - Equival a `true` si el seu **valor numèric és diferent de zero**.
- Seguidament s'aplica la **negació**.
 - **Exemple**: Si `$a = 7`, `$a` equival a `true` i `!$a` és `false`.
 - **Exemple**: Si `$a = 0`, `$a` equival a `false` i `!$a` és `true`.

Operador de negació aplicat sobre nombres o text

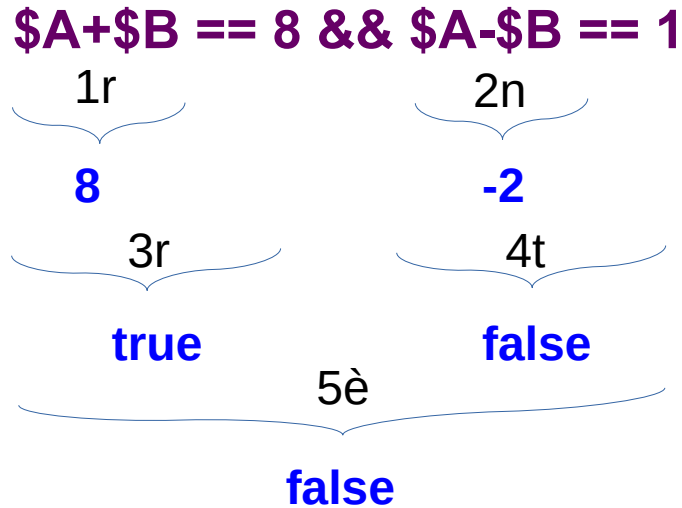
- Per **cadena de text**:
 - La **cadena buida** equival a **false**.
 - **Qualsevol altra cadena** equival a **true**.
- Per **exemple**
 - Si **\$text1 = ""** (cadena buida) aleshores:
 - **\$text1** equival a **false**.
 - **!\$text1** equival a **true**.

Ordre de prioritat,
prelació o precedència

Operadors lògics



Exemple avaluació d'expressions: $\$A=3$ i $\$B=5$



Exemple1.php

```
<?php
    $a = 3;
    $b = 9;
    $resultat = ($a <= 3 and $b <> 9);
    if ($resultat == true) {
        echo "Es compleix la condició";
    } else {
        echo "No es compleix la condició";
    }
?>
```

Comparació de cadenes de text amb operadors relacionals

- Quan comparem dues cadenes de text:
 - Es comparen **lletra a lletra** pel valor numèric (ASCII) de **cada lletra**.
 - Per exemple: la a és el número 97, la b el 98, etc.
 - Si comparem **"avellana" < "mel"** obtenim **true**.
 - Si comparem **"cotxe" > "vehicle"** obtenim **false**.



- Els codis numèrics poden generar **resultats no previstos**.
- Quin codi numèric és menor?
 - El de la **a** o el de la **A**?
 - Si mirem la taula ASCII: (a = 97 i A = 65)
 - Tots els **codis** numèrics ASCII de les lletres **majúscules són menors que els de les lletres minúscules**.
- **Exemple:** 'Zulu' < 'avellaner' retorna **true**.
 - Quan, alfabèticament esperem que siga a l'inrevés.

4) Operadors lògics i de comparació

5) Operadors aritmètics

- Operadors aritmètics

Nom	Exemple	Resultat	Exemple (amb \$a=8 i \$b=4)
Suma	$\$a + \b	El resultat de la suma.	12
Resta	$\$a - \b	El resultat de la resta.	4
Multiplicació	$\$a * \b	El resultat de la multiplicació.	32
Divisió	$\$a / \b	El resultat de la divisió.	2
Resta de la divisió o mòdul	$\$a \% \b	La resta de la divisió de \$a entre \$b	0

- En el cas del mòdul, els números es **converteixen** a **enters abans d'efectuar l'operació**.
- Per **exemple**:
 - **9 % 4.5** dóna com a resultat **1** i no **0**.
 - Calcula la resta de **9** entre **4**, no de **9** entre **4.5**.
 - Perquè realitza l'operació **9 % 4**, en convertir 4.5 a enter.

Operador Resta de la divisió entera o mòdul

- Útil en **processos repetitius** en programació.
- Ús exclusiu entre nombres **enters**.
- Per **exemple**:
 - $7 \% 3 \Rightarrow 1$ (la resta de dividir 7 entre 3 és 1).
 - $8 \% 2 \Rightarrow 0$ (la resta de dividir 8 entre 2 és zero).

\$a	\$a % 3
1	1
2	2
3	0
4	1
5	2
6	0
7	1
8	2

Exponenciació (calcular una potència)

- **Multiplicar n vegades** el terme.
 - Per exemple a^3 la podem calcular com **$a*a*a$** .
 - No és pràctic per a potències d'exponents grans.
- Fer servir un **bucle** que done lloc a la **repetició de l'operació multiplicació** n vegades.
- Utilitzar **eines pròpies del llenguatge** que permeten realitzar aquesta operació.
 - Opció més senzilla: **`pow(base, exponent)`** .
 - Per exemple: **`pow (2, 3)`** retorna 2 elevat a 3 que resulta **8**.

- Ja hem vist que les **expressions amb operadors** segueixen un **ordre de prelació** o de **precedència** que determinen l'**ordre amb què s'executen**.
- Amb **diversos operadors del mateix nivell**:
 - L'operació s'executa **d'esquerra a dreta**.
- **Recomanació**:
 - Per evitar **resultats no desitjats**.
 - Per deixar **clar** amb quin **ordre s'han d'executar** les operacions
 - En casos on puga existir **dubte**:
 - **Ús de parèntesi**.

- Per **exemple**:
 - L'expressió: $3*a/7+2$
 - Queda més clara si fem els parèntesis: $3*((a/7)+2)$.
 - S'executa **del parèntesis més intern, cap enfora**.
 - 1) $(a / 7)$
 - 2) $(a / 7) + 2$
 - 3) $3 * ((a / 7) + 2)$

Operadors d'increment i decrement

- ++ i -- són **només vàlids per a variables numèriques**.
- Serveixen per **incrementar una unitat** el valor de la **variable**.
- Depenent d'**on es col·loquen** (**abans** o **després** de la variable):
 - El **resultat** del càlcul pot diferir:
 - A causa del **moment en què s'executa** l'addició de la unitat.
- Cal tenir en compte que ++, --, +=, -= i *= són expressions que **sempre s'apliquen sobre variables**.
- Per **exemple** : no és vàlid escriure 2++

Operadors d'increment i decrement

Nom	Exemple	Resultat
Pre-increment	<code>++\$a</code>	Incrementa \$a en 1 i després retorna \$a
Post-increment	<code>\$a++</code>	Retorna \$a i després incrementa \$a en 1.
Pre-decrement	<code>--\$a</code>	Decrementa \$a en 1 i després retorna \$a
Post-decrement	<code>\$a--</code>	Retorna \$a i després decrementa \$a en 1.

Codi exemple2.php:

```
<?php
    $a=8 ;
    echo $a++;
    echo "<br>" ;
    echo $a;

?>
```

Post-increment: Mostra:
8
9

Codi exemple3.php:

```
<?php
    $a=8 ;
    echo ++$a;
    echo "<br>" ;
    echo $a;

?>
```

Pre-increment: Mostra:
9
9

Operadors d'assignació (formes abreujades, sempre sobre variables)

Nom	Exemple	Resultat
Suma	<code>\$a += \$b;</code>	<code>\$a = \$a + \$b;</code>
Resta	<code>\$a -= \$b;</code>	<code>\$a = \$a - \$b;</code>
Multiplicació	<code>\$a *= \$b;</code>	<code>\$a = \$a * \$b;</code>
Divisió	<code>\$a /= \$b;</code>	<code>\$a = \$a / \$b;</code>
Resta de la divisió entera o mòdul	<code>\$a %= \$b;</code>	<code>\$a = \$a % \$b;</code>