

# Algorithmique & Programmation

## Notion de condition

[yann.secq@univ-lille.fr](mailto:yann.secq@univ-lille.fr)

ABDELKADER Omar, BIRLOUEZ Martin, BONEVA Iovka, DELECROIX Fabien, LEQUINIOU Erwann, MARSHALL-BRETON Christopher, REKIK Yosra, SECQ Yann, SOW Younoussa, SUDHEENDRAN Megha, SUE Yue

# TRUE or FALSE

Pigs are raised only  
for their meat.

Pigs have bad eyesight.

A pig can have 10-16 piglets.

Pigs are very dirty.

When a pig gives birth  
it is called farrowing.

A pig's gestation period  
is exactly 3 months,  
3 weeks, 3 days.

# Qu'est-ce qu'une condition ?

- Une condition est une proposition qui est **VRAIE** ou **FAUSSE** dans un contexte donné
- Quelques exemples:
  - L'amphi est calme
  - $3 < 4$
  - $X$  est pair
  - Température  $> 30$  degrés

# Opérateurs de comparaison



Strictement supérieur à	>
Supérieur ou égal à	>=
Strictement inférieur à	<
Inférieur ou égal à	<=
Égal à	==
Différent de	!=

**ATTENTION : affectation (=) ≠ égalité (==) !**

# Quelques conditions ...

Expression conditionnelle	Evaluation de l'expression
<code>3 &gt; 4</code>	
<code>x &lt;= 3</code>	
<code>log(15) &gt;= 2</code>	
<code>equals(saisie, "OK")</code>	
<code>length("Titi") &gt; 2</code>	

Expression conditionnelle	Evaluation de l'expression
<code>3 &gt; 4</code>	FAUX
<code>x &lt;= 3</code>	Dépend de x
<code>log(15) &gt;= 2</code>	VRAI
<code>equals(saisie, "OK")</code>	Dépend de saisie
<code>length(« Titit ») &gt; 2</code>	VRAI

G ö d e l

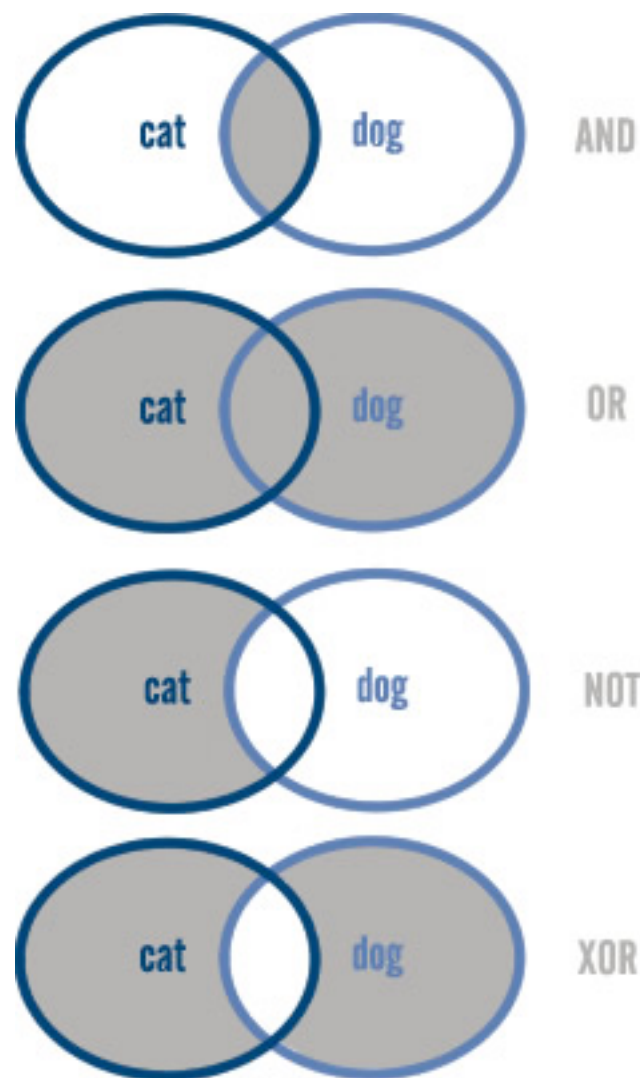
T u r i n g



# Le type Booléen

- Une condition vaut VRAI (`true`) ou FAUX (`false`)
- Le type booléen (`boolean`) représente une condition
- Espace de valeur réduit à {VRAI, FAUX} ou {`true`, `false`}

# Conditions complexes



- Opérateurs logiques:
  - Conjonction: AND / ET / & &
  - Disjonction: OR / OU / | |
  - Négation: NOT / NON / !
  - Disjonction exclusive: XOR / OU EXCLUSIF / ^



# NON: la négation

- (NON a) s'évalue à VRAI si a vaut FAUX et VRAI sinon
- Représente la négation d'une condition

a	NON a (!a)
VRAI	FAUX
<b>FAUX</b>	<b>VRAI</b>

# La conjonction: ET / & &

- $(a \text{ ET } b)$  s'évalue à VRAI si et seulement si  $a$  et  $b$  valent VRAI et FAUX sinon

a	b	a <b>ET</b> b (a&&b)
FAUX	FAUX	FAUX
FAUX	VRAI	FAUX
VRAI	FAUX	FAUX
<b>VRAI</b>	<b>VRAI</b>	<b>VRAI</b>

# La disjonction: $\cup$ / $||$

- $(a \cup b)$  s'évalue à FAUX si et seulement si  $a$  et  $b$  valent FAUX, sinon à VRAI

a	b	$a \cup b$ ( $a    b$ )
FAUX	FAUX	FAUX
<b>FAUX</b>	<b>VRAI</b>	<b>VRAI</b>
<b>VRAI</b>	<b>FAUX</b>	<b>VRAI</b>
<b>VRAI</b>	<b>VRAI</b>	<b>VRAI</b>

# Disjonction exclusive: XOR / $\wedge$

- $(a \text{ XOR } b)$  s'évalue à VRAI si a et b ont des valeurs différentes, sinon à FAUX

a	b	a XOR b ( $a \wedge b$ )
FAUX	FAUX	FAUX
<b>FAUX</b>	<b>VRAI</b>	<b>VRAI</b>
<b>VRAI</b>	<b>FAUX</b>	<b>VRAI</b>
VRAI	VRAI	FAUX

# Echauffement

	a=0	a=11	a=13
	b=-3	b=16	b=16
a > 10			
a <= 12			
(a > 10) ET (a <= 12)			
b > 10			
NON (b > 10)			
((a > 10) ET (a <= 12)) OU (NON(b > 10))			
(a > 12) XOR (b == 0)			

# Echauffement

	a=0	a=11	a=13
	b=-3	b=16	b=16
a > 10	F	V	V
a <= 12	V	V	F
(a > 10) ET (a <= 12)	F	V	F
b > 10	F	V	V
NON (b > 10)	V	F	F
((a > 10) ET (a <= 12)) OU (NON(b > 10))	V	V	F
(a > 12) XOR (b == 0)	F	F	V



# Synthèse sur les types

Type	Java	Espace de valeurs
Entier (8 bits)	<code>byte</code>	$[-128, +127]$
Entier (16 bits)	<code>short</code>	$[-32768, +32767]$
Entier (32 bits)	<b><code>int</code></b>	$[-2,147,483,648, +2,147,483,647]$
Entier (64 bits)	<code>long</code>	$[-9,223,372,036,854,775,808, +9,223,372,036,854,775,807]$
Réel (32 bits)	<code>float</code>	$[2^{-149}, (2-2^{-23}) \cdot 2^{127}]$
Réel (64 bits)	<b><code>double</code></b>	$[2^{-1074}, (2-2^{-52}) \cdot 2^{1023}]$
Booléen	<b><code>boolean</code></b>	<code>{true, false}</code>
Caractère	<b><code>char</code></b>	<code>'.'</code> (Tous les symboles unicode)
Chaîne	<b><code>String</code></b>	<code>"..."</code>

