



أنواع البيانات والعمليات



أنواع البيانات

النوع	الحجم Bits	نوع القيمة	القيمة	مثال
byte	8	عدد صحيح	-128 إلى 127	byte b = 100;
char	16	عدد صحيح	-32768 إلى 32767	char c = 'A';
short	16	عدد صحيح	-32768 إلى 32767	short s = 552;
int	32	عدد صحيح	-2147483648 إلى 2147483647	int i = 88231;
long	64	عدد صحيح	-9223372036854775808 إلى 9223372036854775807	long l = 22137424;
float	32	كسور عشرية	-3.40282347E+38F إلى 3.40282347E+38F	float f = 2.31;
double	64	كسور عشرية	-1.80E+308 إلى 1.80E+308	double d = 3.4153255;
boolean	-	قيمة بولية\منطقية	-	boolean b = true;
void	-	-	-	void func();



العمليات الرياضية Arithmetic Operations

العملية	الوصف	مثال (A=10, B=20)
+ الجمع	جمع القيمتين اللتين تقعان على جانبي المعامل +	A + B يساوي 30
- الطرح	طرح القيمة على يمين المعامل - من القيمة التي تقع على يسار المعامل -	A - B يساوي -10
* الضرب	ضرب القيمتين اللتين تقعان على جانبي المعامل *	A * B يساوي 200
/ القسمة	تقسيم القيمة التي تقع على يسار المعامل / على القيمة التي تقع على اليمين المعامل /	A / B يساوي 2
% باقي القسمة	تقسيم القيمة على اليسار على القيمة على اليمين وارجاع قيمة باقي القسمة	A % B يساوي 0
++ الزيادة	زيادة القيمة بمقدار 1 صحيح	++B يساوي 21
-- التقيص	تنقيص القيمة بمقدار 1 صحيح	--B يساوي 19



عمليات العلاقات Conditional Operations

العملية	الوصف	مثال (A=10, B=20)
== يساوي	يتحقق الشرط ويصبح صحيحا إذا تساوت القيمتان	A == B غير صحيح
!= لا يساوي	يتحقق الشرط ويصبح صحيحا إذا لم تتساوى القيمتين	A != B صحيح
> أكبر من	يتحقق الشرط ويصبح صحيحا إذا كانت القيمة على اليسار أكبر من القيمة على اليمين	A > B غير صحيح
< أصغر من	يتحقق الشرط ويصبح صحيحا إذا كانت القيمة على اليسار أصغر من القيمة على اليمين	A < B صحيح
>= أكبر من أو يساوي	يتحقق الشرط ويصبح صحيحا إذا كانت القيمة على اليسار أكبر من أو تساوي القيمة على اليمين	A >= B غير صحيح
<= أصغر من أو يساوي	يتحقق الشرط ويصبح صحيحا إذا كانت القيمة على اليسار أصغر من أو تساوي القيمة على اليمين	A <= B صحيح



العمليات المنطقية Logical Operations

العملية	الوصف	مثال (A = true, B = false)
(و) &&	تسمى بـ Logical AND والتي يكون ناتجها True في حال كان كل من القيمتين على جانبيها true	A && B يساوي false
(أو)	وتسمى بـ Logical OR والتي يكون ناتجها True في حال كان أي من القيمتين على جانبيها true	A B يساوي true
(لا) !	تسمى بـ logical NOT ووظيفتها نفي القيمة المنطقية ففي حال كانت القيمة true سيكون الناتج false والعكس صحيح	!(A && B) يساوي true



عمليات التعيين Assignment Operations

العملية	الوصف	مثال (A=10, B=20)
المساواة =	معامل المساواة الأبسط والذي من خلاله بإعطاء القيمة على اليمين للمتغير الموجود على اليسار	$C = A + B$ ستكون قيمة C هنا تساوي 30
الإضافة +=	معامل الإضافة والمساواة ووظيفته جمع القيمة على اليمين مع المتغير على اليسار وتخزين الناتج في المتغير على اليسار	$C = C + A$ تكافئ $C += A$
الطرح -=	معامل الطرح والمساواة ووظيفته طرح القيمة على اليمين من قيمة المتغير على اليسار وتخزين الناتج في المتغير على اليسار	$C = C - A$ تكافئ $C -= A$
الضرب *=	معامل الضرب والمساواة والذي يقوم بضرب القيمتين على اليمين وعلى اليسار ثم تخزين الناتج في المتغير على اليسار	$C = C * A$ تكافئ $C *= A$
القسمة /=	معامل التقسيم والمساواة والذي يقوم بتقسيم اليسار على اليمين ثم تخزين الناتج في المتغير على اليسار	$C = C / A$ تكافئ $C /= A$
باقي القسمة %=	معامل باقي القسمة والمساواة والذي يقوم بتخزين باقي قسمة اليسار على اليمين في المتغير على اليسار	$C = C \% A$ تكافئ $C \% = A$



أولوية العمليات Operations Precedence

الأولوية	العملية/العامل	التحرك حسب الأولوية
1	[] . ()	من اليسار إلى اليمين
2	i++ i--	-
3	++i --i ! ~	من اليمين إلى اليسار
4	new	من اليمين إلى اليسار
5	* / %	من اليسار إلى اليمين
6	- +	من اليسار إلى اليمين
7	<< >> >>>	من اليسار إلى اليمين
8	< <= > >=	-
9	== !=	من اليسار إلى اليمين
10	&	من اليسار إلى اليمين
11	^	من اليسار إلى اليمين
12		من اليسار إلى اليمين
13	&&	من اليسار إلى اليمين
14		من اليسار إلى اليمين
15	?:	من اليمين إلى اليسار
16	= += -= *= /= %= &= ^= = <<= >>= >>>=	من اليمين إلى اليسار