Programmazione I

Lezione 6

Il sistema dei tipi

- Tipi “base” in GO
  - `bool`
  - `string`
  - `int  int8  int16  int32  int64`
  - `uint uint8 uint16 uint32 uint64 uintptr`
  - `byte // alias for uint8`
  - `rune // alias for int32, represents a Unicode code point`
  - `float32 float64`
  - `complex64 complex128`
- int e uint sono “implementation dependent” (32 o 64 bit)

Il sistema dei tipi

- Tipi “base” in GO
  - `bool`
  - `string`
  - **`int  int8  int16  int32  int64`**
  - **`uint uint8 uint16 uint32 uint64`** `uintptr`
  - `byte // alias for uint8`
  - `rune // alias for int32, represents a Unicode code point`
  - `float32 float64`
  - `complex64 complex128`
- int e uint sono “implementation dependent” (32 o 64 bit)

Operatori in GO

Binari						
*	/	%	<<	>>	&	&^
+	-	|	^			
==	!=	<	<=	>	>=	
&&						
||						
Unari						
++	--					
+	-	!	^	*	&	<-

Operatori in GO

Binari						
*	/	%	<<	>>	&	&^
+	-	|	^			
==	!=	<	<=	>	>=	
&&						
||						
Unari						
++	--					
+	-	!	^	*	&	<-

Il sistema dei tipi

- Tipi “base” in GO
  - bool
  - string
  - int int8 int16 int32 int64
  - uint uint8 uint16 uint32 uint64 uintptr
  - byte // alias for uint8
  - rune // alias for int32, represents a Unicode code point
  - **float32 float64**
  - complex64 complex128

Operatori in GO

Binari						
*	/	%	<<	>>	&	&^
+	-	|	^			
==	!=	<	<=	>	>=	
&&						
||						
Unari						
++	--					
+	-	!	^	*	&	<-

Strong typing in GO

- Esperimenti sulla filosofia “strong typing” di GO
- Il meccanismo di conversione temporanea di tipo (***cast e coercion)***

Le costanti

- Idea:
  - valori (literals) con particolare significato
  - Ricorrono piu` volte nel programma
  - Esempio: valore di π
  - Alcune già *predefinite* in GO (esempio math.Pi, math.MaxInt32)
- Esempio: dichiarazione di costanti in GO