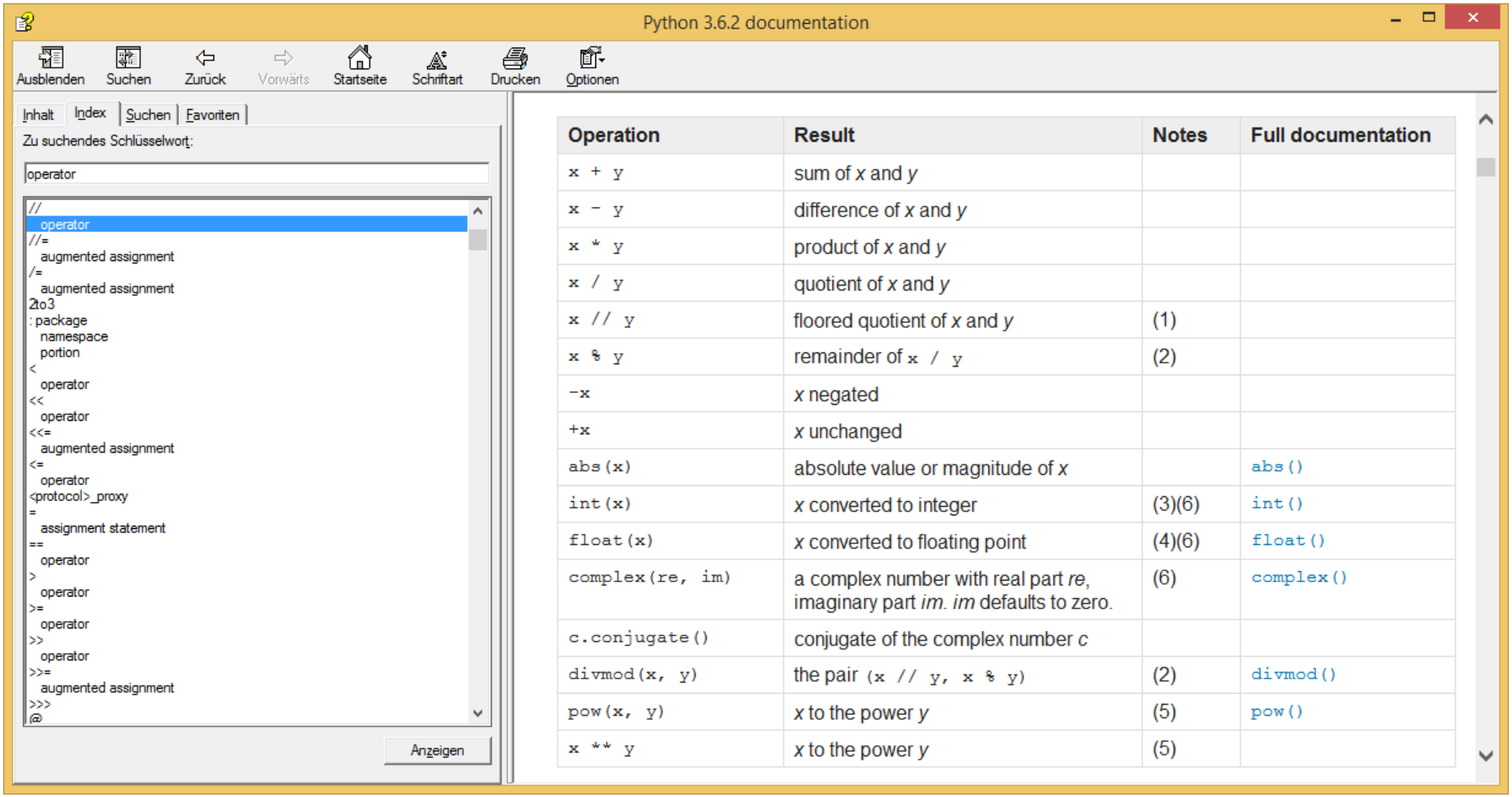
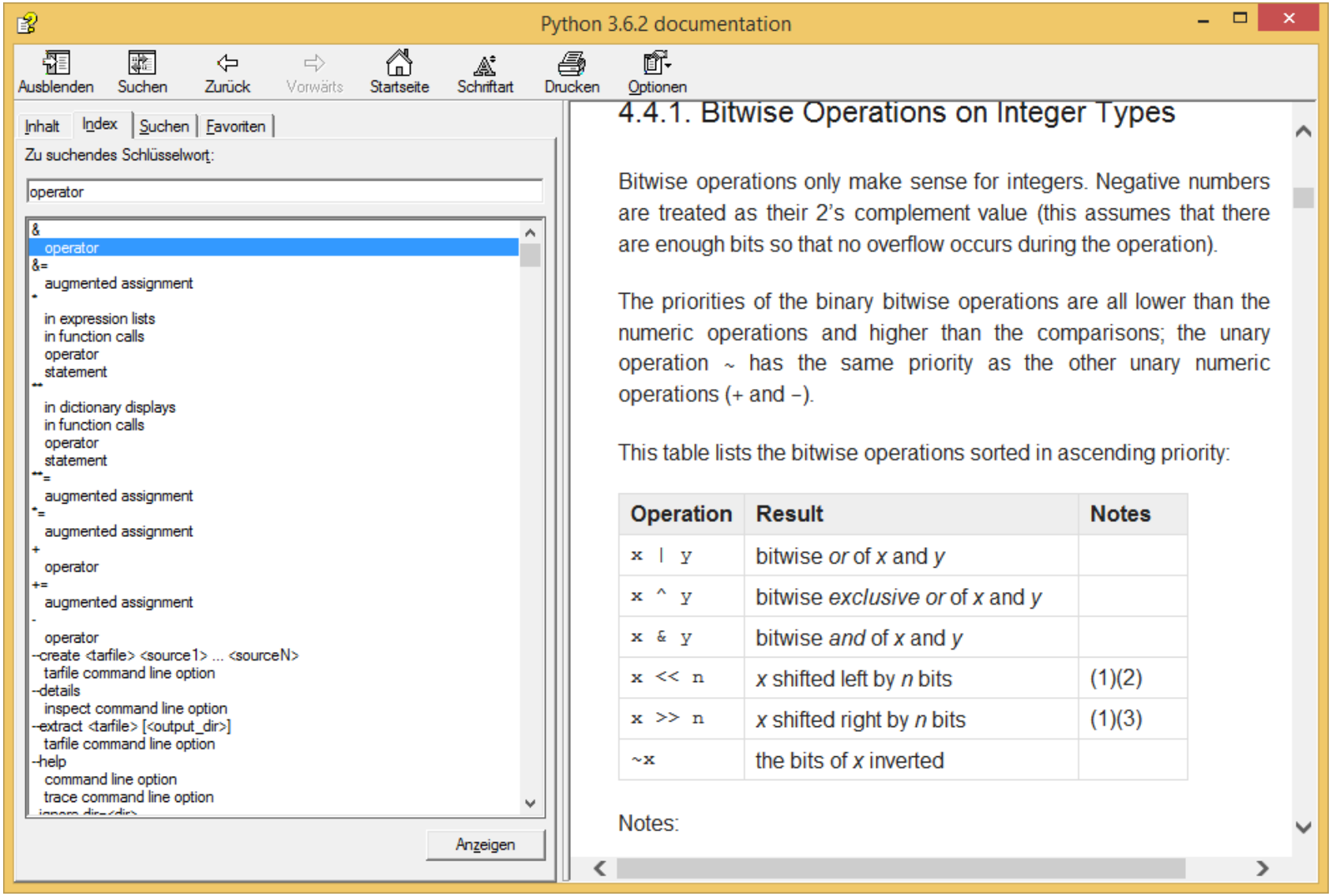
1.3 Interaktive Nutzung: Python Shell



Operation	Result	Notes	Full documentation
`x + y`	sum of *x* and *y*		
`x - y`	difference of *x* and *y*		
`x * y`	product of *x* and *y*		
`x / y`	quotient of *x* and *y*		
`x // y`	floored quotient of *x* and *y*	(1)	
`x % y`	remainder of `x / y`	(2)	
`-x`	*x* negated		
`+x`	*x* unchanged		
`abs(x)`	absolute value or magnitude of *x*		`abs()`
`int(x)`	*x* converted to integer	(3)(6)	`int()`
`float(x)`	*x* converted to floating point	(4)(6)	`float()`
`complex(re, im)`	a complex number with real part *re*, imaginary part *im*. *im* defaults to zero.	(6)	`complex()`
`c.conjugate()`	conjugate of the complex number *c*		
`divmod(x, y)`	the pair `(x // y, x % y)`	(2)	`divmod()`
`pow(x, y)`	*x* to the power *y*	(5)	`pow()`
`x ** y`	*x* to the power *y*	(5)	

1.3 Interaktive Nutzung: Python Shell



1.3 Interaktive Nutzung: Python Shell – Bitweises Beispiel

Bit Position	Bits 4	3	2	1	0	-1	-2	-3
Berechnung Wert	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
Wert	16	8	4	2	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$
Beispiel 9	0	1	0	0	1			
(9 << 1) = 18	1	0	0	1	0			
(18 >> 2) = 4	0	0	1	0	0			
6	0	0	1	1	0			
4 & 6 = 4	0	0	1	0	0			
4 | 6 = 6	0	0	1	1	0			
4 ^ 6 = 2	0	0	0	1	0			
4 ^ 6 ^ 6 = 4	0	0	1	0	0			

Bitweise Verknüpfungen

UND

&	0	1
0	0	0
1	0	1

ODER

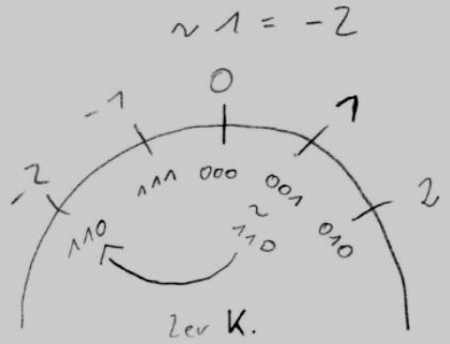
|	0	1
0	0	1
1	1	1

XOR

^	0	1
0	0	1
1	1	0

NOT

~	
0	1
1	0



1.3. interaktives Beispiel, Erklärung 2er Komplement