Binding Strengths

For two entities X and Y that are adjacent in an expression (that is, XY), the binding strength between them and the result of the bind is shown in this table:

		Y						
		A	**F**	**H**	**MOP**	**DOP**	**DOT**	**IDX**
X	**A**	6 A	3 AF	3 AF	4 F		7 REF	4 A
	F	2 A	1 F	4 F	4 F			4 F
	H		1 F	4 F	4 F			4 H
	AF	2 A	1 F					
	MOP			4 ERR				
	DOP	5 MOP	5 MOP	5 MOP				
	JOT	5 MOP	5 MOP	5 MOP	4 F			
	DOT	6 ERR	5 MOP	5 MOP		6 ERR		
	REF	7 A	7 F	7 H	7 MOP	7 DOP		
	IDX	3 ERR	3 ERR	3 ERR				

where:

A : *Array, for example, `0 1 2 'hello' ⍺ ⍵`
F : *Function (primitive/defined/derived/system), for example, `+ - +.× myfn ⎕CR {⍺ ⍵}`
H : *Hybrid function/operator, that is, `/ ⌿ \ ⍀`
AF : Bound left argument, for example, `2+`
MOP : *Monadic operator, for example, `¨ ⍨ &`
DOP : Dyadic operator, for example, `⍣ ⌸ ⍤ ⌺`
JOT : Jot, that is, compose/null operand `∘`
DOT : Dot, that is, reference/product `.`
IDX : square-bracketed expression, for example, `[⍺+⍳⍵]`
ERR : Error

* indicates a "first-class" entity, which can be parenthesised or named

In this table:

- the higher the number, the stronger the binding
- an empty field indicates no binding for this combination; an error.

For example, in the expression `a b.c[d]`, where `a`, `b`, `c` and `d` are arrays, the binding proceeds:

```
      a b . c [d]
       6 7 6 4      ⍝ binding strengths between entities

→     a (b.) c [d]
       0    7 4

→     a (b.c) [d]
       6     4

→     (a(b.c))[d]
```

Note: Binding strengths can vary between APL distributions. For example, the expression `a b c[d]` binds as `(a b c)[d]` in Dyalog (and other pre-nested array APLs such as APL\360) rather than as `a b (c[d])`, which is preferred by APL2 and APLX (among others).