

True or false? (unless otherwise specified.) Assume any explicitly named package is installed. An expression that raises an error does not evaluate to `true`.

Answer grid

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
True	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
False	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

- The expression `"ab" * "cd" * "!"` returns the string `"abcd!"`.
- The following code produces a single plot containing both curves, rather than replacing the first curve with the second.

```
using Plots
plot(x -> x^2, xlims=(-1, 1), label="square")
plot!(x -> x^3, label="cube")
```

- The expression `eps()/2 + eps()/2 == eps()` evaluates to `true`.
- The expression `typeof(1 + 2.0) == Float64` evaluates to `true`.
- If `u = [1, 2]` and `v = [3, 4]`, then `[u v]` returns the matrix `[1 2; 3 4]`.
- If `A = [1 2 3; 4 5 6]`, then `A[2, 3]` returns 6.

- The following code assigns the vector `[2, 5]` to `w`.

```
v = [-3, 2, 0, 5, -1]
w = v[v .> 0]
```

- The following code assigns the vector `[1, 3, 5]` to `a`.

```
f(x, y) = x - 2y
a = f.([3, 5, 7], 1)
```

- The expression `sum(length, ["Julia", "is", "fun"])` returns 3.
- The command `A = [i + 2j for i in 1:2, j in 1:3]` produces the matrix `A = [3 5 7; 4 6 8]`.
- After the following definitions, `f(f(1))` returns 2.0.

```
f(x::Float64) = 1
f(x::Int) = 2.0
```

12. The call `odd_sum(5)` returns 9.

```
function odd_sum(n)
    s = 0
    for k in 1:n
        if isodd(k)
            s += k
        end
    end
    return s
end
```

13. The expression `LinRange(0, 1, 4) == [0, 0.2, 0.4, 0.6, 0.8, 1]` evaluates to `true`.

14. With the following definition, the expression `f(n)` returns the n -th Fibonacci number for positive integers n (assuming no integer overflow). Recall that the Fibonacci numbers are defined as $f_1 = f_2 = 1$ and $f_k = f_{k-1} + f_{k-2}$ for $k \geq 3$.

```
f(n) = n <= 2 ? 1 : f(n-1) + f(n-2)
```

15. If `v = [2, 4, 6, 8]`, then `v[2:end] == [2, 4, 6]` evaluates to `true`.

Bonus (2 points). Give the values of `a` and `b` after the following code, and briefly explain why the results differ.

```
v = [[1, 2, 3], [2, 3, 4], [4, 5, 6]]
a = sum.(v)
b = sum(v)
```

Answer and explanation: