DATA ANALYSIS - THE DATA TABLE WAY

INDEXING

Using column names in i

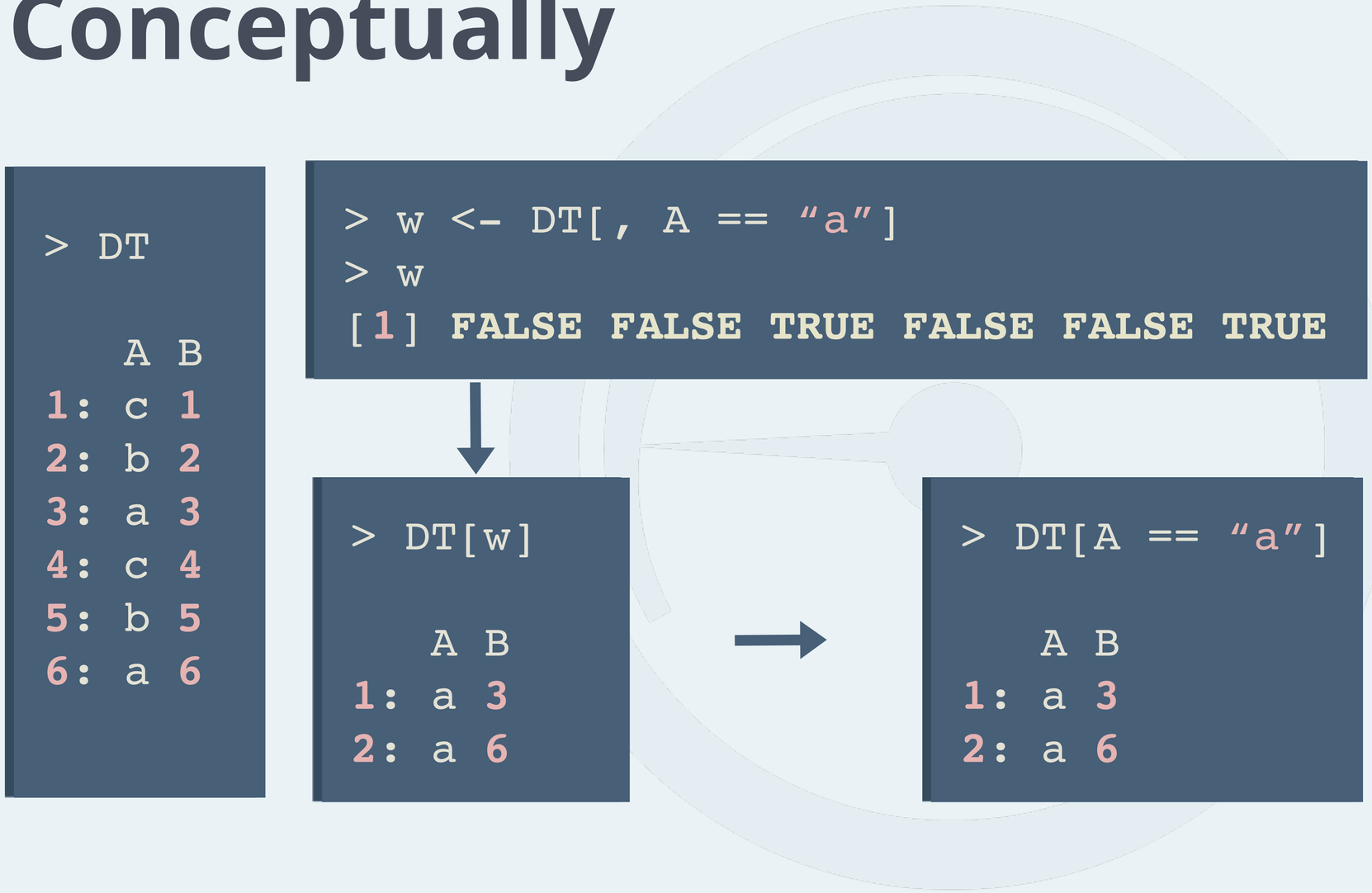
DATA.TABLE

```
> DT
    A B
1: c 1
2: b 2
3: a 3
4: c 4
5: b 5
6: a 6
```

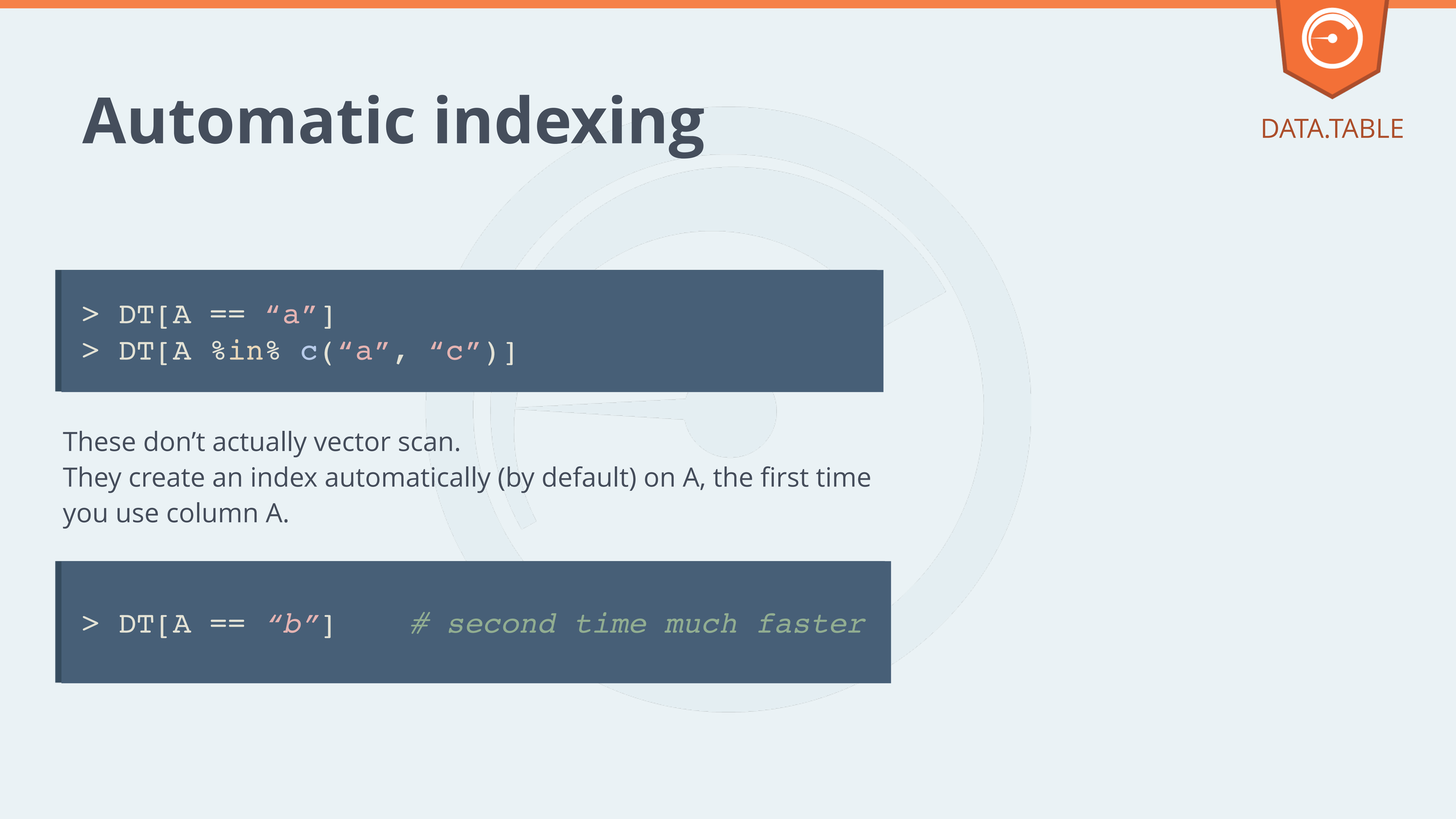
```
> DT[A == “a”]

    A B
1: a 3
2: a 6
```

```
> DT[A %in% c(“a”, “c”)]

    A B
1: c 1
2: a 3
3: c 4
4: a 6
```

Conceptually



Automatic indexing




```
> DT[A == “a”]
> DT[A %in% c(“a”, “c”)]
```

These don’t actually vector scan.
They create an index automatically (by default) on A, the first time you use column A.

```
> DT[A == “b”]     # second time much faster
```

DATA.TABLE

Let’s practice

DATA ANALYSIS - THE DATA TABLE WAY

KEYS

Creating and using a key


```
> DT

   A B
1: c 1
2: b 2
3: a 3
4: c 4
5: b 5
6: a 6
```

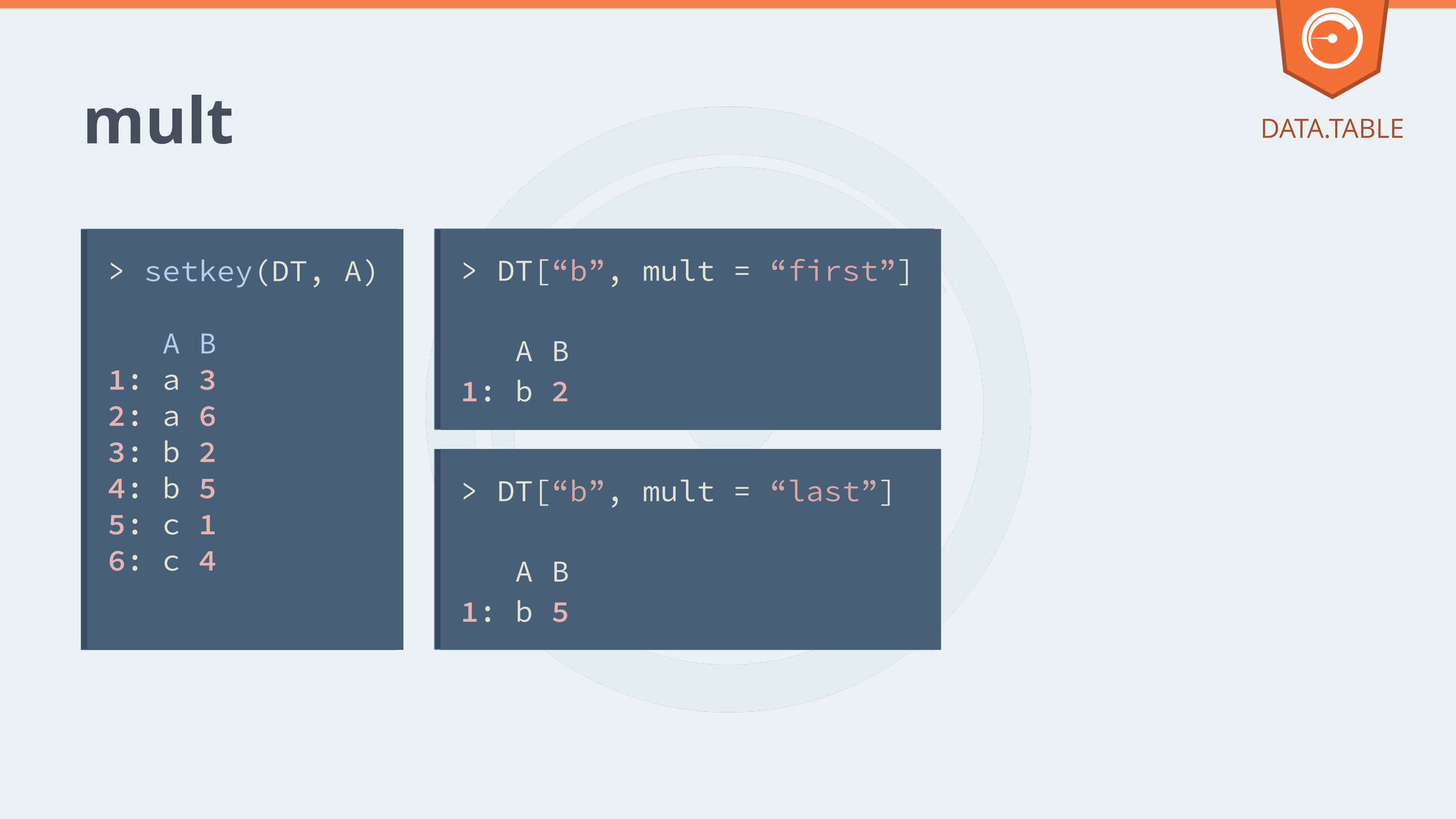
```
> setkey(DT, A)

   A B
1: a 3
2: a 6
3: b 2
4: b 5
5: c 1
6: c 4
```

```
> DT

   A B
1: a 3
2: a 6
3: b 2
4: b 5
5: c 1
6: c 4
```

```
> DT[“b”]

   A B
1: b 2
2: b 5
```

```
> DF[“b”]
Error:
duplicate 'row.names'
are not allowed
```

mult




```
> setkey(DT, A)

   A B
1: a 3
2: a 6
3: b 2
4: b 5
5: c 1
6: c 4
```

```
> DT["b", mult = "first"]

   A B
1: b 2
```

```
> DT["b", mult = "last"]

   A B
1: b 5
```

nomatch

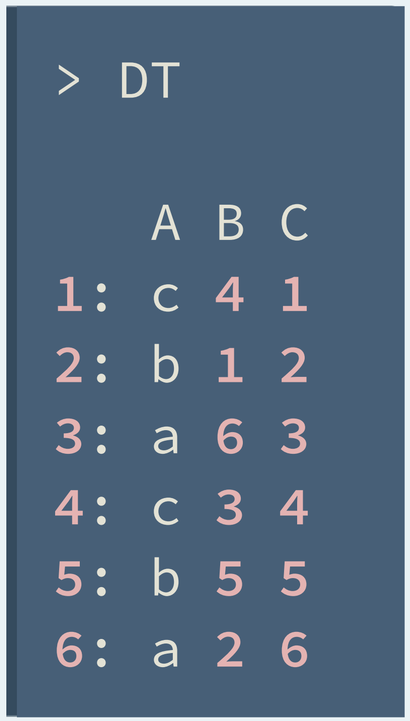
DATA.TABLE

```
> setkey(DT, A)

   A B
1: a 3
2: a 6
3: b 2
4: b 5
5: c 1
6: c 4
```

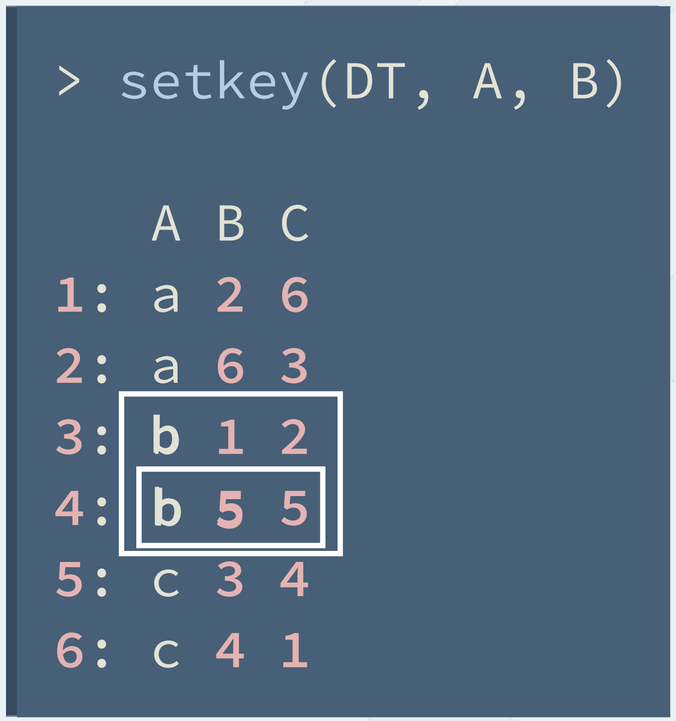
```
> DT[c("b", "d"), nomatch = NA] # default

   A B
1: b 2
2: b 5
3: d NA
```

```
> DT[c("b", "d"), nomatch = 0]

   A B
1: b 2
2: b 5
```

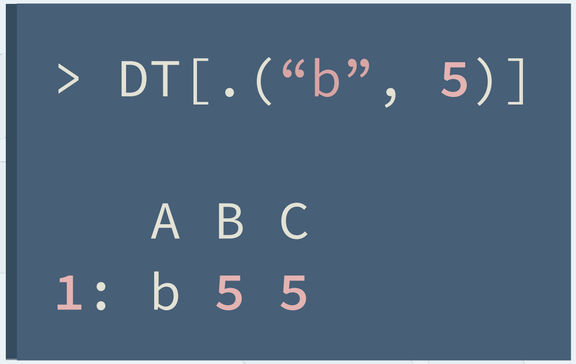
A two-column key



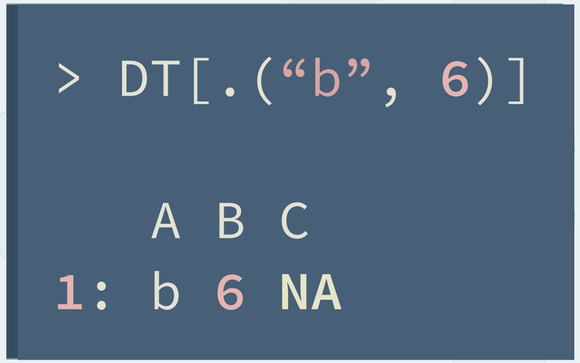
```
> DT

   A B C
1: c 4 1
2: b 1 2
3: a 6 3
4: c 3 4
5: b 5 5
6: a 2 6
```



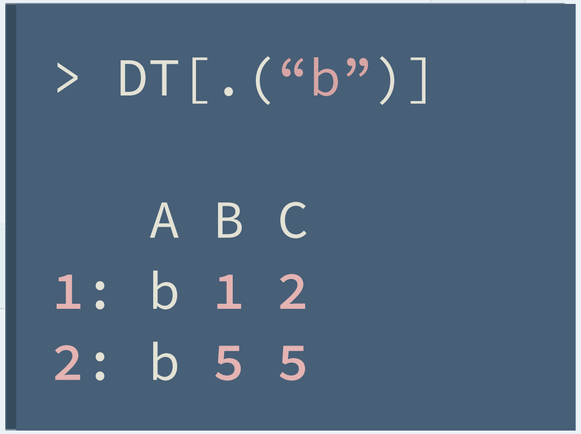
```
> setkey(DT, A, B)

   A B C
1: a 2 6
2: a 6 3
3: b 1 2
4: b 5 5
5: c 3 4
6: c 4 1
```



```
> DT[.("b", 5)]

   A B C
1: b 5 5
```



```
> DT[.("b", 6)]

   A B C
1: b 6 NA
```



```
> DT[.("b")]

   A B C
1: b 1 2
2: b 5 5
```

DATA.TABLE

DATA.TABLE

Let’s practice

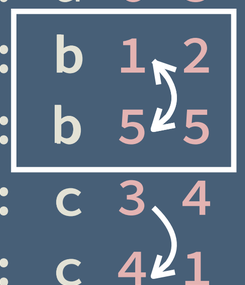
DATA ANALYSIS - THE DATA TABLE WAY

ROLLING JOINS

Ordered joins

DATA.TABLE

```
> setkey(DT, A, B)

   A B C
1: a 2 6
2: a 6 3
3: b 1 2
4: b 5 5
5: c 3 4
6: c 4 1
```



```
> DT[.("b", 4)]

   A B C
1: b 4 NA
```

```
> DT[.("b", 4), roll = TRUE]

   A B C
1: b 4 2
```

```
> DT[.("b", 4), roll = "nearest"]

   A B C
1: b 4 5
```

Forwards and backwards

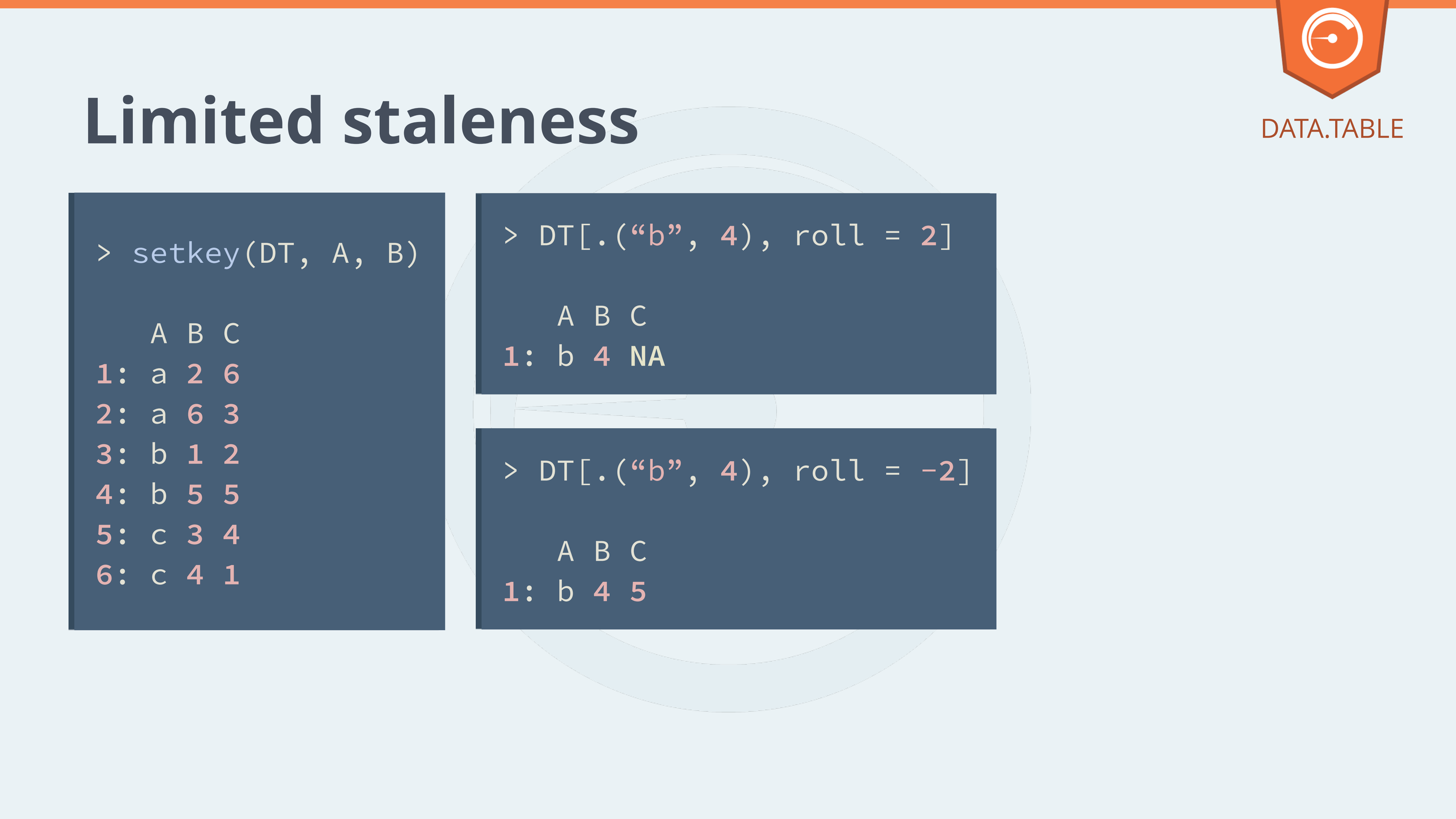
DATA.TABLE

```
> setkey(DT, A, B)

   A B C
1: a 2 6
2: a 6 3
3: b 1 2
4: b 5 5
5: c 3 4
6: c 4 1
```

```
> DT[.("b", 4), roll = +Inf]

   A B C
1: b 4 2
```

```
> DT[.("b", 4), roll = -Inf]

   A B C
1: b 4 5
```

Limited staleness



DATA.TABLE

```
> setkey(DT, A, B)

   A B C
1: a 2 6
2: a 6 3
3: b 1 2
4: b 5 5
5: c 3 4
6: c 4 1
```

```
> DT[.("b", 4), roll = 2]

   A B C
1: b 4 NA
```

```
> DT[.("b", 4), roll = -2]

   A B C
1: b 4 5
```

Control ends

DATA.TABLE

```
> setkey(DT, A, B)

   A B C
1: a 2 6
2: a 6 3
3: b 1 2
4: b 5 5
5: c 3 4
6: c 4 1
```

```
> DT[.("b", 7:8), roll = TRUE]

   A B C
1: b 7 5
2: b 8 5
```

```
> DT[.("b", 7:8), roll = TRUE, rollends = FALSE]

   A B  C
1: b 7 NA
2: b 8 NA
```

DATA.TABLE

Let’s practice