🦓 Zebraw

Zebraw is a lightweight and fast package for displaying code blocks with line numbers in typst, supporting code line highlighting. The term "**Zebraw**" is a combination of "**zebra**" and "**raw**", for the highlighted lines will be displayed in the code block like a zebra lines.

Example

To use, import `zebraw` package then follow with `#show zebraw.with()` .

```
#import "@preview/zebraw:0.3.0": *

#show: zebraw.with()

```typ
hi
It's a raw block with line numbers.
```
```

```
hi
It's a raw block with line numbers.
```

The line spacing can be adjusted by passing the `inset` parameter to the `zebraw` function. The default value is `top: 3pt, bottom: 3pt, left: 3pt, right: 3pt`. Notice that by using `#show` , the `inset` parameter will be applied to all code blocks except code blocks rendered by `#zebraw()` function.

```
#show: zebraw.with(
  inset: (top: 6pt, bottom: 6pt)
)

```typ
hi
It's a raw block with line numbers.
```
```

```
hi
It's a raw block with line numbers.
```

For cases where code line highlighting is needed, you should use `#zebraw()` function with `highlight-lines` parameter to specify the line numbers that need to be highlighted, as shown below:

```
#zebraw(
  highlight-lines: (1, 3),
  ```typ
  It's me,
  hi!
  I'm the problem it's me.
  ```,
)
```

```
It's me,
hi!
I'm the problem it's me.
```

Customize the highlight color by passing the `highlight-color` parameter to the `zebraw` function:

```
#zebraw(
  highlight-lines: (1,),
  highlight-color: blue.lighten(90%),
  ```typ
  I'm so blue!
              -- George III
  ```,
)
```

```
I'm so blue!
             -- George III
```

For more complex highlighting, you can also add comments to the highlighted lines by passing an array of line numbers and comments to the `highlight-lines` parameter:

````
#zebraw(
  highlight-lines: (
    (1, "auto indent!"),
    (2, [Content available as *well*.]),
    3
  ),
  highlight-color: blue.lighten(90%),
  comment-font-args: (
    fill: blue,
    font: "IBM Plex Sans"
  ),
  comment-flag: "⤳",
  ```typ
  I'm so blue!
               -- George III
  I'm not.
               -- Alexander Hamilton
  ```,
)
````

````
1 I'm so blue!
⤳ auto indent!
2              -- George III
               ⤳ Content available as well.
3 I'm not.
4              -- Alexander Hamilton
````

You can also add a header or footer to the code block by passing the `header` / `footer` parameter to the `zebraw` function, as shown below:

````
#zebraw(
  lang: true,
  header: "this is the example of the header",
  ```typ
  I'm so blue!
               -- George III
  I'm not.
               -- Alexander Hamilton
  ```,
  footer: "this is the end of the code",
)
````

````
this is the example of the header                typst
1 I'm so blue!
2              -- George III
3 I'm not.
4              -- Alexander Hamilton
this is the end of the code
````

To change the rendered results of both pure typst raw block and `zebraw` block, you can use the `zebraw-init` function to set the default values for `highlight-color`, `inset`, `comment-color`, `comment-flag`, `comment-font-args`, and `lang`:

```
#show: zebraw-init.with(
  highlight-color: rgb("#94e2d5").lighten(50%),
  inset: (top: 0.3em, right: 0.3em, bottom: 0.3em, left: 0.3em),
  comment-color: none,
  comment-flag: ">",
  comment-font-args: (size: 8pt),
```

```
  lang: true,
)
```

Without using `zebraw-init`, you can still begin with just `zebraw` function and use the default values. By using `zebraw-init` without any parameters, the values will be reset to the default values.

Real-world Example

Here is an example of using `zebraw` to highlight lines in a Rust code block:

Calculate Fibonacci number using reccursive function — rust

```rust
pub fn fibonacci_reccursive(n: i32) -> u64 {
    if n < 0 {
        panic!("{} is negative!", n);
    > to avoid negative numbers
    }
    match n {
        0 => panic!("zero is not a right argument to fibonacci_reccursive()!"),
        1 | 2 => 1,
        3 => 2,
        _ => fibonacci_reccursive(n - 1) + fibonacci_reccursive(n - 2),
    > /*
      50 => 12586269025,
      */
    }
}
```