

- 1. 计算  $x + y$
- 2. 第一布里渊区的范围是：  $-\frac{\pi}{a} < k < \frac{\pi}{a}$
- 3. 白日依山尽， 黄河入海流

问题：

- 如果传入的公式是行间公式，不清楚如何放置下划线

$$\underline{f(x) dx}$$

- 不清楚如何针对公式的宽度自适应下划线的位置

对于 p<sup>+</sup>n 结，其扩散电容的表达式为： $C_D = \left(\frac{Aq^2p_{n0}L_p}{k_oT}\right) \exp\left(\frac{qV}{k_oT}\right)$

你好。下面是一段测试文字。

```
1 x = 3
2 print("Hello, World!")
```

Python

Here is the content of the variable **with style**

```
Hello World
1 #let forecast(day) = block[
2   #box(square(
3     width: 2cm,
4     inset: 8pt(c),
5     fill: if day(a).weather == "sunny"(b) {
6       yellow
7     } else {
8       aqua
9     },
10    align(
11      bottom + right,
12      strong(day.weather),
13    ),
14  ))
15  #h(6pt)
16  #set text(22pt, baseline: -8pt)
17  #day.temperature °#day.unit // 你好呀
18 ]
19 #let forecast(day) = block[
```

Typst

```
Hello World
20 #box(square(
21   width: 2cm,
22   inset: 8pt,
23   fill: if day.weather == "sunny" {
24     yellow
25   } else {
26     aqua
27   },
28   align(
29     bottom + right,
30     strong(day.weather),
31   ),
32 ))
33 #h(6pt)
34 #set text(22pt, baseline: -8pt)
35 #day.temperature °#day.unit
36 ]
```

Typst

zebraw 测试

Adding **rbx** to **rcx** gives the desired result.

What is **fn main()** in Rust would be **int main()** in C.

```
1 fn main() {
2   println!("Hello World!");
3 }
```

Rust

This has **`backticks`** in it (but the spaces are trimmed). And **here** the leading space is also trimmed.

**Hello world**

**Hello World**

10

raw(text: "strong", block: false, lang: "typc")

**content**

Hello world ! this is a very very very very very very very very very very lo

**ong text**