


```
1 pub fn main() {
2     println!("Hello, world!");
3 }
```

 Rust

```
1 def fibonacci(n):
2     if n <= 1:
3         return n
4     else:
5         return(fibonacci(n-1) + fibonacci(n-2))
```

 Python

We can also set a line number offset with `codly-offset(int)`:


```
2     println!("Hello, world!");
```

 Rust

We are also able to control line numbers alignment:


`#codly(numbers-alignment: horizon)`

```
1 import numpy as np
   print(np.array([np.random.randint(1, 100) for _ in range(1000)]),
2 np.array([np.random.normal(0, 1) for _ in range(1000)]),
   np.array([np.random.uniform(0, 1) for _ in range(1000)]))
```

 Python


`#codly(numbers-alignment: top)`

```
1 import numpy as np
2 print(np.array([np.random.randint(1, 100) for _ in range(1000)]),
   np.array([np.random.normal(0, 1) for _ in range(1000)]),
   np.array([np.random.uniform(0, 1) for _ in range(1000)]))
```

 Python


And we can also disable line numbers:

```
pub fn main() {
    println!("Hello, world!");
}
```

 Rust

We can also select only a range of lines to show:

```
return(fibonacci(n-1) + fibonacci(n-2))
```

 Python

```
pub fn main() {
    println!("Hello, world!");
}
```

 Rust

```
1 pub fn main() {
2     println!("Hello, world!");
3 }
```

 Rust

```
1 pub fn function<R, S, T>() -> R where T: From<S>, S: Into<R>, R: Send +
   Sync + 'static {
2     println!("Hello, world!");
3 }
```

 Rust

```
1 pub fn main() {  
2     println!("This is in another page!")  
3 }
```

Rust

```
1 pub fn main() {  
2     println!("Strong line numbers go brrrrrrr.");  
3 }
```

Rust