

Bitcoin SegWit address (bech32)

bc1... Bitcoin SegWit address.

```
^(bc1)[02-9ac-hj-np-z]{38,87}$
```

Token-by-token breakdown

Token	Meaning
^	start of string (or line in multiline mode)
(	start of a capturing group
bc1	literal text "bc1"
)	end of group
[02-9ac-hj-np-z]{38,87}	between 38 and 87 times: any of: "0", 2–9, "a", c–h, j–n, p–z
\$	end of string (or line in multiline mode)

Quick usage in 7 languages

JavaScript `const re = /^(bc1)[02-9ac-hj-np-z]{38,87}$/;
re.test(value);`

Python `import re
re.match(r"^(bc1)[02-9ac-hj-np-z]{38,87}$", value)`

Java `Pattern.compile("^(bc1)[02-9ac-hj-np-z]{38,87}$").matcher(value).matches();`

C# / .NET `Regex.IsMatch(value, @"^(bc1)[02-9ac-hj-np-z]{38,87}$");`

Go `regexp.MustCompile(`^(bc1)[02-9ac-hj-np-z]{38,87}$`).MatchString(value)`

Ruby `/^(bc1)[02-9ac-hj-np-z]{38,87}$/.match?(value)`

PHP `preg_match('~^(bc1)[02-9ac-hj-np-z]{38,87}$~', $value);`

Common pitfalls

- **Anchored to the whole string.** This pattern uses ^ and \$, so it requires the entire input to match. To find it inside a longer text, drop the anchors and use the global (g) flag.
- **Case sensitivity.** Letter ranges are case-sensitive — use the i flag if the input case can vary.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the bitcoin segwit address (bech32) against a source of truth (database, API, or checksum) where it matters.