Literal characters - it will find exactly the one that we specified, for example, if we specify car, it will find exactly car and not cars.

Dot - The point will find us the symbol that we indicate, that is, it depends on the points that we specified, if we specify r"m.p" it will find map, map, map there should be only one symbol, it will not find maap, moup.

Caret - The caret will find the word but it must be at the beginning for example if r"Car" it will find "Car is blue" and not "Blue Car".

Dollar - Our dollar works the other way around for caret, that is, it looks not at the beginning, but at the very back.

Square brackets - Using brackets, we can search for any letter that is inside the bracket, for example [f, s, d] allows us to find any letter from s, d, f.

Hyphen - It turns out that we are specifying a range here and it will search for characters that are in our range, for example r"[a-z]" search for all small letters

Asterisk - this function tells us that this character can be repeated any number of times, even 0 times. For example r”[c\*r]” will find car, car, cr.

Plus - And here we need to have at least 1 character, for example, r"[c+r]" will find car, caar but not cr.

Question Mark - And here the symbol must appear once or not at all, for example r"[c?r]" will find car or cr.

Parentheses - And parentheses will help us group parts together, for example “(cr)+” would be crcrcrcr.

Pipe - this means "or" that is, it will find at least 1 of our specified ones, for example, r"(home|car)" - will find "home" or "car".

Backflash - and we will insert this so that the element becomes normal, for example, r"\." – will find the dot "."