

Credit Card 16 Digit Validation

Yuani Chen

Did You Know?

There are many security functions built into your credit card. One of which includes a numbering system for the 16 digits of your credit card.

It appears that the 16th digit of your credit card number is a result of the 1st 15 digits!

The Math Behind It

The calculation of the 16th digit utilises the Luhn algorithm.

The Luhn algorithm or Luhn formula, also known as the "modulus 10" or "mod 10" algorithm, is a simple checksum formula used to validate a variety of identification numbers, such as credit card numbers, IMEI numbers, National Provider Identifier numbers in the US, and Canadian Social Insurance Numbers.

Source: Wikipedia

Using the App

First the user enters the 1st 15 digits of his/her credit card number in the field provided.

```
textInput("cc_number15","Enter the 1st 15 Digits of the Credit Card No. :",value="")
```

<!--html_preserve-->

Enter the 1st 15 Digits of the Credit Card No. :

<!--/html_preserve-->

Then click on the button to find your 16th digit!

```
actionButton("checkButton","Find the 16th Digit!")
```

<!--html_preserve--> <!--/html_preserve-->

Result! - Checking input values

The action button triggers the calculation on the server side.

First, we check that exactly 15 digits were entered into the field.

```
output$len15<-renderText({ if (str_length(input$cc_number15)!=15)
  {"Please enter 15 digits ONLY!"
  }
  else
  {return("You have entered 15 digits.")
  }
})
```

Results! - Final calculation

On condition that 15 digits have been entered, we then evaluate the 16th digit. Here is a snapshot of the final calculation. We return d16 as the resultant value in the UI.

```
d16 <- if(str_length(input$cc_number15)==15)

{
  if ((round_any(d16_pre,10)-d16_pre)>=0)
  {round_any(d16_pre,10)-d16_pre} else
  {round_any(d16_pre,10)+10-d16_pre}
}
else
{return("")}
```