

CS 649 Big Data: Tools and Methods  
Fall Semester, 2022  
Doc 14 Memoization  
Feb 22, 2022

Copyright ©, All rights reserved. 2022 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/openpub/>) license defines the copyright on this document.

# Memoization

Storing the results of expensive function calls

Returning the cached result when the same inputs occur again

# Manual Memoization

```
function factorial (n is a non-negative integer)
  if n is 0 then
    return 1 [by the convention that 0! = 1]
  else if n is in lookup-table then
    return lookup-table-value-for-n
  else
    let x = factorial(n - 1) times n [recursively invoke factorial
      with the parameter 1 less than n]
    store x in lookup-table in the nth slot [remember the result of n! for later]
    return x
  end if
end function
```

# Automatic Memoization

```
import streamlit as st
```

```
@st.experimental_memo  
def factorial(n):  
    if n < 1:  
        return 1  
    return n * factorial(n - 1)
```

```
f20 = factorial(200)  
st.write(factorial(180))
```

```
@st.cache  
def factorial(n):  
    if n < 1:  
        return 1  
    return n * factorial(n - 1)
```