


```

# Find the a element with class="a-link-normal s-no-outline" and get the link
a_element <- html_node(div_element, 'a.a-link-normal.s-no-outline')
link <- ifelse(!is.na(a_element), paste0("https://amazon.com", html_attr(a_element, "href")), '')

# Find the img element with class="s-image" and get the link
img_element <- html_node(div_element, 'img.s-image')
img_src <- ifelse(!is.na(img_element), html_attr(img_element, "src"), '')

# Find the span element with class="a-size-base-plus a-color-base a-text-normal" and get the title
title_element <- html_node(div_element, 'span.a-size-base-plus.a-color-base.a-text-normal')
title <- ifelse(!is.na(title_element), html_text(title_element), '')

# Find the span element with class="a-price-whole" and get the price
price_element <- html_node(div_element, 'span.a-price-whole')
price <- ifelse(!is.na(price_element), html_text(price_element), '')

# Find the span element with class="a-icon-alt" and get the ratings
rating_element <- html_node(div_element, 'span.a-icon-alt')
rating <- ifelse(!is.na(rating_element), html_text(rating_element), '')

# Append data to vectors
links <- c(links, link)
img_srcs <- c(img_srcs, img_src)
titles <- c(titles, title)
prices <- c(prices, price)
ratings <- c(ratings, rating)
}

# Create a data frame with the scraped data
product_df <- data.frame(
  Links = links,
  Images = img_srcs,
  Title = titles,
  Price = prices,
  Rating = ratings
)

# Write the data to a CSV file
write.csv(product_df, "WatchMeWhip.csv", row.names = FALSE)

#catSupplies
# Install and load the rvest package
#if (!requireNamespace("rvest", quietly = TRUE)) {
#   install.packages("rvest")
#}

## Warning: package 'rvest' is in use and will not be installed

#}
library(rvest)
library(polite)
# Read the HTML file
url <- "https://www.amazon.com/s?i=specialty-aps&bbn=16225013011&rh=n%3A%2116225013011%2Cn%3A2975241011"

```

```
session <- bow(url, user_agent = "Student's Demo Educational")
session
```

```
## <polite session> https://www.amazon.com/s?i=specialty-aps&bbn=16225013011&rh=n%3A%2116225013011%2Cn%
## User-agent: Student's Demo Educational
## robots.txt: 137 rules are defined for 4 bots
## Crawl delay: 5 sec
## The path is scrapable for this user-agent
```

```
session_page <- scrape(session)
```

```
# Find all div elements with the specified class
```

```
div_elements <- html_nodes(session_page, 'div.sg-col-4-of-24.sg-col-4-of-12.s-result-item.s-asin.sg-col-
```

```
# Create empty vectors to store data
```

```
links <- character()
```

```
img_srcs <- character()
```

```
titles <- character()
```

```
prices <- character()
```

```
ratings <- character()
```

```
max_products <- 29
```

```
# Limit the loop to only collect data for the first 30 products
```

```
for (i in 1:min(length(div_elements), max_products)) {
```

```
  div_element <- div_elements[i]
```

```
# Find the a element with class="a-link-normal s-no-outline" and get the link
```

```
a_element <- html_node(div_element, 'a.a-link-normal.s-no-outline')
```

```
link <- ifelse(!is.na(a_element), paste0("https://amazon.com", html_attr(a_element, "href")), '')
```

```
# Find the img element with class="s-image" and get the link
```

```
img_element <- html_node(div_element, 'img.s-image')
```

```
img_src <- ifelse(!is.na(img_element), html_attr(img_element, "src"), '')
```

```
# Find the span element with class="a-size-base-plus a-color-base a-text-normal" and get the title
```

```
title_element <- html_node(div_element, 'span.a-size-base-plus.a-color-base.a-text-normal')
```

```
title <- ifelse(!is.na(title_element), html_text(title_element), '')
```

```
# Find the span element with class="a-price-whole" and get the price
```

```
price_element <- html_node(div_element, 'span.a-price-whole')
```

```
price <- ifelse(!is.na(price_element), html_text(price_element), '')
```

```
# Find the span element with class="a-icon-alt" and get the ratings
```

```
rating_element <- html_node(div_element, 'span.a-icon-alt')
```

```
rating <- ifelse(!is.na(rating_element), html_text(rating_element), '')
```

```
# Append data to vectors
```

```
links <- c(links, link)
```

```
img_srcs <- c(img_srcs, img_src)
```

```
titles <- c(titles, title)
```

```
prices <- c(prices, price)
```

```
ratings <- c(ratings, rating)
```

```

}

# Create a data frame with the scraped data
product_df <- data.frame(
  Links = links,
  Images = img_srcs,
  Title = titles,
  Price = prices,
  Rating = ratings
)

# Write the data to a CSV file
write.csv(product_df, "CatSupplies.csv", row.names = FALSE)

```

```

#Furniture
# Install and load the rvest package
# if (!requireNamespace("rvest", quietly = TRUE)) {
  install.packages("rvest")
}

```

```
## Warning: package 'rvest' is in use and will not be installed
```

```

#}
library(rvest)
library(polite)
# Read the HTML file
url <- "https://www.amazon.com/s?i=specialty-aps&bbn=16225011011&rh=n%3A%2116225011011%2Cn%3A1063306&ref=sr_pg_1"

session <- bow(url, user_agent = "Student's Demo Educational")
session

```

```

## <polite session> https://www.amazon.com/s?i=specialty-aps&bbn=16225011011&rh=n%3A%2116225011011%2Cn%3A1063306&ref=sr_pg_1
##   User-agent: Student's Demo Educational
##   robots.txt: 137 rules are defined for 4 bots
##   Crawl delay: 5 sec
##   The path is scrapable for this user-agent

```

```

session_page <- scrape(session)

# Find all div elements with the specified class
div_elements <- html_nodes(session_page, 'div.sg-col-4-of-24.sg-col-4-of-12.s-result-item.s-asin.sg-col-4-of-24')

# Create empty vectors to store data
links <- character()
img_srcs <- character()
titles <- character()
prices <- character()
ratings <- character()

max_products <- 29

# Limit the loop to only collect data for the first 30 products
for (i in 1:min(length(div_elements), max_products)) {

```

```

div_element <- div_elements[i]

# Find the a element with class="a-link-normal s-no-outline" and get the link
a_element <- html_node(div_element, 'a.a-link-normal.s-no-outline')
link <- ifelse(!is.na(a_element), paste0("https://amazon.com", html_attr(a_element, "href")), '')

# Find the img element with class="s-image" and get the link
img_element <- html_node(div_element, 'img.s-image')
img_src <- ifelse(!is.na(img_element), html_attr(img_element, "src"), '')

# Find the span element with class="a-size-base-plus a-color-base a-text-normal" and get the title
title_element <- html_node(div_element, 'span.a-size-base-plus.a-color-base.a-text-normal')
title <- ifelse(!is.na(title_element), html_text(title_element), '')

# Find the span element with class="a-price-whole" and get the price
price_element <- html_node(div_element, 'span.a-price-whole')
price <- ifelse(!is.na(price_element), html_text(price_element), '')

# Find the span element with class="a-icon-alt" and get the ratings
rating_element <- html_node(div_element, 'span.a-icon-alt')
rating <- ifelse(!is.na(rating_element), html_text(rating_element), '')

# Append data to vectors
links <- c(links, link)
img_srcs <- c(img_srcs, img_src)
titles <- c(titles, title)
prices <- c(prices, price)
ratings <- c(ratings, rating)
}

# Create a data frame with the scraped data
product_df <- data.frame(
  Links = links,
  Images = img_srcs,
  Title = titles,
  Price = prices,
  Rating = ratings
)

# Write the data to a CSV file
write.csv(product_df, "Furniture.csv", row.names = FALSE)

#CellphoneAccessories
# Install and load the rvest package
#if (!requireNamespace("rvest", quietly = TRUE)) {
  install.packages("rvest")
}

## Warning: package 'rvest' is in use and will not be installed

#}
library(rvest)
library(polite)
# Read the HTML file

```

```
url <- "https://www.amazon.com/s?i=specialty-aps&bbn=16225009011&rh=n%3A%2116225009011%2Cn%3A2811119011"

session <- bow(url, user_agent = "Student's Demo Educational")
session
```

```
session_page <- scrape(session)

# Find all div elements with the specified class
div_elements <- html_nodes(session_page, 'div.sg-col-4-of-24.sg-col-4-of-12.s-result-item.s-asin.sg-col-4-of-24')

# Create empty vectors to store data
links <- character()
img_srcs <- character()
titles <- character()
prices <- character()
ratings <- character()

max_products <- 29

# Limit the loop to only collect data for the first 30 products
for (i in 1:min(length(div_elements), max_products)) {
  div_element <- div_elements[i]

  # Find the a element with class="a-link-normal s-no-outline" and get the link
  a_element <- html_node(div_element, 'a.a-link-normal.s-no-outline')
  link <- ifelse(!is.na(a_element), paste0("https://amazon.com", html_attr(a_element, "href")), '')

  # Find the img element with class="s-image" and get the link
  img_element <- html_node(div_element, 'img.s-image')
  img_src <- ifelse(!is.na(img_element), html_attr(img_element, "src"), '')

  # Find the span element with class="a-size-base-plus a-color-base a-text-normal" and get the title
  title_element <- html_node(div_element, 'span.a-size-base-plus.a-color-base.a-text-normal')
  title <- ifelse(!is.na(title_element), html_text(title_element), '')

  # Find the span element with class="a-price-whole" and get the price
  price_element <- html_node(div_element, 'span.a-price-whole')
  price <- ifelse(!is.na(price_element), html_text(price_element), '')

  # Find the span element with class="a-icon-alt" and get the ratings
  rating_element <- html_node(div_element, 'span.a-icon-alt')
  rating <- ifelse(!is.na(rating_element), html_text(rating_element), '')

  # Append data to vectors
  links <- c(links, link)
  img_srcs <- c(img_srcs, img_src)
  titles <- c(titles, title)
}
```

```

prices <- c(prices, price)
ratings <- c(ratings, rating)
}

# Create a data frame with the scraped data
product_df <- data.frame(
  Links = links,
  Images = img_srcs,
  Title = titles,
  Price = prices,
  Rating = ratings
)

# Write the data to a CSV file
write.csv(product_df, "CellphonesAccessories.csv", row.names = FALSE)

```

```

#Sports&Fitness
# Install and load the rvest package
#if (!requireNamespace("rvest", quietly = TRUE)) {
  install.packages("rvest")
}

```

```
## Warning: package 'rvest' is in use and will not be installed
```

```

#}
library(rvest)
library(polite)
# Read the HTML file
url <- "https://www.amazon.com/s?i=specialty-aps&bbn=16225014011&rh=n%3A%2116225014011%2Cn%3A1097118101"

session <- bow(url, user_agent = "Student's Demo Educational")
session

```

```

## <polite session> https://www.amazon.com/s?i=specialty-aps&bbn=16225014011&rh=n%3A%2116225014011%2Cn%
##   User-agent: Student's Demo Educational
##   robots.txt: 137 rules are defined for 4 bots
##   Crawl delay: 5 sec
##   The path is scrapable for this user-agent

```

```

session_page <- scrape(session)

# Find all div elements with the specified class
div_elements <- html_nodes(session_page, 'div.sg-col-4-of-24.sg-col-4-of-12.s-result-item.s-asin.sg-col-4-of-24')

# Create empty vectors to store data
links <- character()
img_srcs <- character()
titles <- character()
prices <- character()
ratings <- character()

max_products <- 29

```

```

# Limit the loop to only collect data for the first 30 products
for (i in 1:min(length(div_elements), max_products)) {
  div_element <- div_elements[i]

  # Find the a element with class="a-link-normal s-no-outline" and get the link
  a_element <- html_node(div_element, 'a.a-link-normal.s-no-outline')
  link <- ifelse(!is.na(a_element), paste0("https://amazon.com", html_attr(a_element, "href")), '')

  # Find the img element with class="s-image" and get the link
  img_element <- html_node(div_element, 'img.s-image')
  img_src <- ifelse(!is.na(img_element), html_attr(img_element, "src"), '')

  # Find the span element with class="a-size-base-plus a-color-base a-text-normal" and get the title
  title_element <- html_node(div_element, 'span.a-size-base-plus.a-color-base.a-text-normal')
  title <- ifelse(!is.na(title_element), html_text(title_element), '')

  # Find the span element with class="a-price-whole" and get the price
  price_element <- html_node(div_element, 'span.a-price-whole')
  price <- ifelse(!is.na(price_element), html_text(price_element), '')

  # Find the span element with class="a-icon-alt" and get the ratings
  rating_element <- html_node(div_element, 'span.a-icon-alt')
  rating <- ifelse(!is.na(rating_element), html_text(rating_element), '')

  # Append data to vectors
  links <- c(links, link)
  img_srcs <- c(img_srcs, img_src)
  titles <- c(titles, title)
  prices <- c(prices, price)
  ratings <- c(ratings, rating)
}

# Create a data frame with the scraped data
product_df <- data.frame(
  Links = links,
  Images = img_srcs,
  Title = titles,
  Price = prices,
  Rating = ratings
)

# Write the data to a CSV file
write.csv(product_df, "SportsFitness.csv", row.names = FALSE)

```