An eXtensible Business Reporting Language (XBRL) Learning Guide for Business Undergraduates

Vincent Castellani
Pennsylvania State University
Smeal College of Business

March 2023

First Draft – Subject to Change

What Is This Guide?

The process of learning about as well as using XBRL is confusing and very likely to be overwhelming for beginners, especially undergraduate business students. In my belief, this is likely due to the vast amount of relatively unfamiliar terminology used (e.g., taxonomies, tags, metadata) to describe XRBL coupled with very few free informative sources written in plain English. In this guide, I intend to provide new users with a very high-level overview of what XBRL is, how public issuers comply with the SEC's required use of XBRL, and, perhaps most importantly, XBRL's API. The goal of this guide is to allow readers to quickly understand what XBRL is and ultimately use XBRL's API to obtain and analyze data from public issuers' filings. The guide is written with an intended audience of an undergraduate data analytics class and should take no more than a week or two to cover. Finally, to be clear, this guide does not create new material, but simply synthesizes existing sources of data, which are cited throughout, into a simplified user-friendly format.

The exercises covered can be downloaded from the following site - https://github.com/vcastellani/XBRL.

Introduction to XBRL

What is XBRL?

The eXtensible Business Reporting Language (XBRL) is an open international framework for exchanging and analyzing financial data. The XBRL language is managed and published by XBRL International, a non-for-profit global consortium. XBRL is a member of the family of eXtensible Markup Language (XML) languages, which were ultimately designed for the electronic communication of data. XBRL, in particular, was developed for the unique requirements that are demanded of business and financial information. The language was developed to modernize how financial data is reported and shared, addressing the need for a more standardized format that could be easily read and interpreted by both humans and computers. The language's overarching goal is to make financial reporting more accurate, reliable, and accessible. By using a standardized format, the language ensures that financial data from different sources can be compared and analyzed efficiently. XBRL is currently used by more than 600 private and public member organizations in more than 50 countries. XBRL data is currently consumed by regulators, companies, governmental agencies, data providers, analysts, investors, and accountants.

XBRL Advantages

XBRL offers users several advantages when compared to traditional text based reporting including standardization, automation, accuracy, and interoperability, among others. More specifically, XBRL provides a uniform method to describe financial data, which is essential for comparison and analysis (*standardization*). XBRL facilitates automated data extraction and analysis, which reduces the time and cost associated with manual data handling (*automation*). XBRL also helps to reduce the risk of errors that occur in manual processes (*accuracy*). Finally, it enables seamless data exchange across various software, platforms, and organizations (*interoperability*).

XBRL Essentials and Examples

The current XBRL standard is the *XBRL 2.1 Specification* which was originally released in 2003. This specification defines the basic building blocks of the language, which primary include (i) instance documents, (ii) facts, (iii) concepts, and (iv) taxonomies. These four elements are common to any sort of XBRL implementation. Other relevant terms include tags, items, domains, domain members, axes, dimensions, elements, schema, linkbases, and attributes.

Within a financial reporting framework using XBRL, the taxonomy is basically a dictionary of financial reporting terms which defines specific tags for individual items of data, known as facts, such as 'revenue' or 'cost of goods sold'. The instance document is where the actual financial data of a company's filing, for example, is stored. It is essentially a business report in a machine-readable format. Every fact within the instance document is assigned a tag from the taxonomy. The schema is what defines the relationship between the tags and their attributes. Linkbases provide additional context about the relationships between different elements in a taxonomy, enhancing the data's dimensionality and richness.

Detailed definitions of these essential items are defined below…

I. **Instance Documents** – an instance document is a collection of facts that make up a business report. At a technical level, an instance document is an XML document containing a document tree which possesses a root element of <xbrl> </xbrl>.

The Securities and Exchange Commission (hereby, SEC) defines an instance document as an XML file that contains business reporting information and represents a collection of financial facts and report-specific information using tags from one or more XBRL taxonomies.[1]

II. **Facts** – an individual piece of information in a report. At a technical level, facts are represented by elements (see definition below) within the instance document.

The SEC defines a fact as the occurrence in an instance document of a value or other information tagged by a taxonomy element.

III. **Concept** – a definition that provides the meaning of a fact. At a more technical level, concepts correspond to element definitions in an XML Schema.

The SEC defines a concept as the XBRL technical term for an element (see definition below).

IV. **Taxonomy** – a collection of concept definitions. A taxonomy corresponds to a specific reporting regime. For example, taxonomies exist for accounting standards, for government agencies, and for large enterprises. Each taxonomy will consist of an XML Schema document containing element definitions and a collection of XML documents (linkbases) that provide additional information that form the concept definitions.

The SEC defines a taxonomy as an electronic dictionary of business reporting elements used to report business data. A taxonomy is composed of an elements names file (.xsd) and relationships files directly referenced by that schema. The taxonomy schema files together with the relationships files define the concepts (elements) and relationships that form the basis of the taxonomy. The set of related schema and relationships files altogether constitute a taxonomy. The US GAAP Taxonomy used for SEC filers can be found at this link – XBRL Taxonomies. Once loaded, you must select the relevant taxonomy to explore (e.g., US GAAP (2023)).

Other information terminology, as defined by the SEC, include…

Tags – the identifying information that describes a unit of data in an instance document and encloses it in angle brackets (i.e., < >). All facts in an instance document are enclosed by tags that identify the element of the fact.

Items – the XBRL technical term for a kind of element.

Domains – an element that represents an entire set of other elements; the domain and its members are used to classify facts along the axis of a table. For example, 'cost of goods sold' is a domain member in the domain 'expenses'.

Domain Members – an element representing one of the possibilities within a domain.

[1] As covered in more detail in a latter section of this guide, the SEC began requiring public companies to provide their registration documents and periodic and current reports using XBRL. As such, this guide heavily focuses on SEC-relevant XBRL documents; however, it is important to clarify that this is just one application of XBRL. XBRL is used by a number of regulatory agencies, governments, companies not just in America but throughout the world.

Axes –differentiates facts with each specific axis representing a unique way that facts may be classified. For example, ‘revenue’ may be reported along a business unit axis, a country axis, a state axis, an industry axis, a product axis, and so forth.

Dimensions – the XBRL technical term for axis.

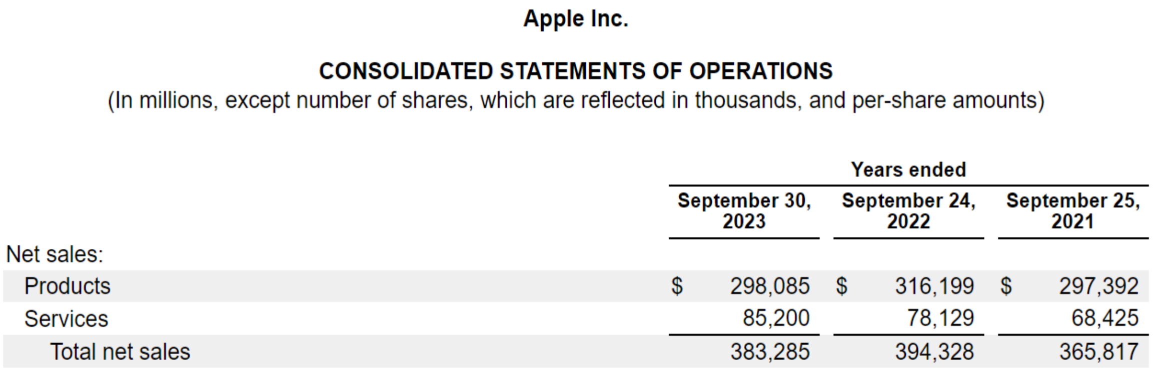
Elements – the XBRL components (items, domain members, dimensions, and so forth). The representation of a financial reporting concept, including line items in the face of the financial statements, important narrative disclosures, and rows and columns in tables.

Schema – the technical term for an element declaration or names file. Here is a link – Apple Inc.'s 2023 10-K Schema Document – to the schema document associated with Apple Inc.’s 2023 10-K filing.

Linkbase – XBRL technical term for a relationships file. There are several linkbases associated with an XBRL filing (e.g., calculation, definition, label, and presentation linkbases).

Attribute – a property of an element such as its name, balance, data type, and whether the element is abstract. Please note that attributes of the XBRL US GAAP Taxonomy elements cannot be changed.

Here are some examples. First, this link – Apple Inc.'s 2023 10-K Instance Document – takes you to the *instance document* associated with Apple Inc.’s 2023 10-K filing. All the *facts* from Apple’s 2023 10-K filing, which was filed as a traditional .hml file, are found within the instance document. For example, the following image is taken from the statement of operations:

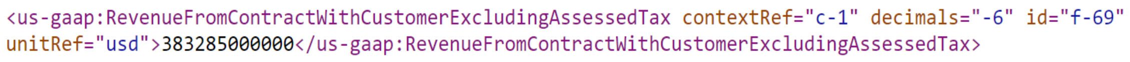


Apple Inc.

CONSOLIDATED STATEMENTS OF OPERATIONS
(In millions, except number of shares, which are reflected in thousands, and per-share amounts)

	Years ended		
	September 30, 2023	**September 24, 2022**	**September 25, 2021**
Net sales:			
Products	$ 298,085	$ 316,199	$ 297,392
Services	85,200	78,129	68,425
Total net sales	383,285	394,328	365,817

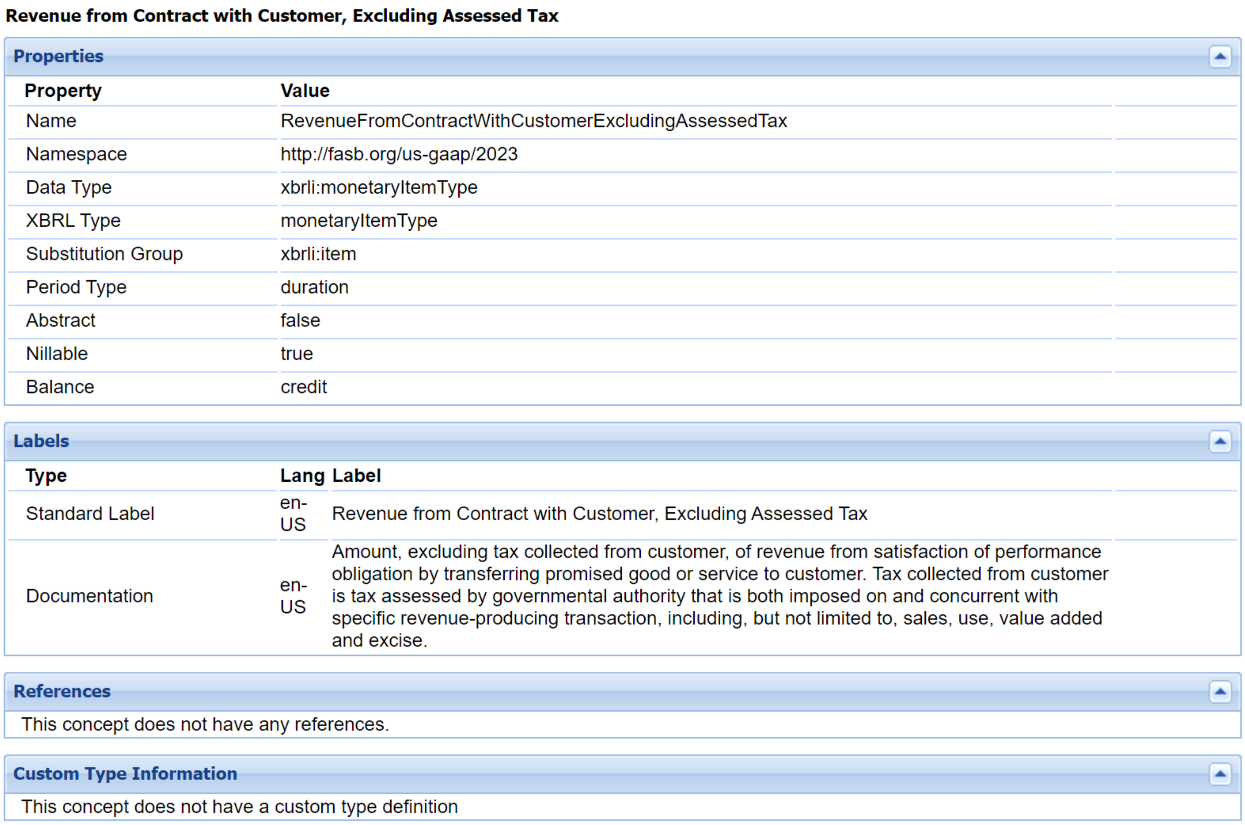
As you can see Apple reported Total Net Sales of $383,285 million for the year ended on September 30th, 2023. If you search for this *fact* within the *instance document*, you will find the corresponding *fact* with the following *attributes*:



```
<us-gaap:RevenueFromContractWithCustomerExcludingAssessedTax contextRef="c-1" decimals="-6" id="f-69"
unitRef="usd">383285000000</us-gaap:RevenueFromContractWithCustomerExcludingAssessedTax>
```

First, this fact is associated with the reporting tag or *concept* of ‘us-gaap:RevenueFromContractWithCustomer Excluding AssessedTax’.[2] The first part identifies the *taxonomy* used (i.e., US GAAP) and the second part identifies the *concept* – Revenue from Contract with Customer, Excluding Assessed Tax. This *concept*, as defined in the *taxonomy*, is displayed below.

[2] More information on this reporting tag can be found by searching the US GAAP taxonomy including properties, labels, references, relationships, and tree locations. The taxonomy can be found using the following link – XBRL Taxonomies. Once loaded, you must select the relevant taxonomy to explore (e.g., US GAAP (2023)).



Revenue from Contract with Customer, Excluding Assessed Tax

Properties

Property	Value
Name	RevenueFromContractWithCustomerExcludingAssessedTax
Namespace	http://fasb.org/us-gaap/2023
Data Type	xbrli:monetaryItemType
XBRL Type	monetaryItemType
Substitution Group	xbrli:item
Period Type	duration
Abstract	false
Nillable	true
Balance	credit

Labels

Type	Lang	Label
Standard Label	en-US	Revenue from Contract with Customer, Excluding Assessed Tax
Documentation	en-US	Amount, excluding tax collected from customer, of revenue from satisfaction of performance obligation by transferring promised good or service to customer. Tax collected from customer is tax assessed by governmental authority that is both imposed on and concurrent with specific revenue-producing transaction, including, but not limited to, sales, use, value added and excise.

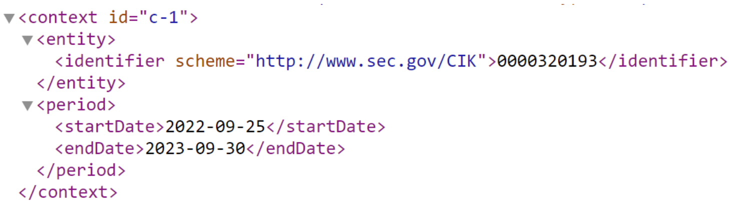
References

This concept does not have any references.

Custom Type Information

This concept does not have a custom type definition

The *fact* is further associated with the context reference (contextRef) of ‘c-1’. If you search for this within the *instance document*, you will find the following:



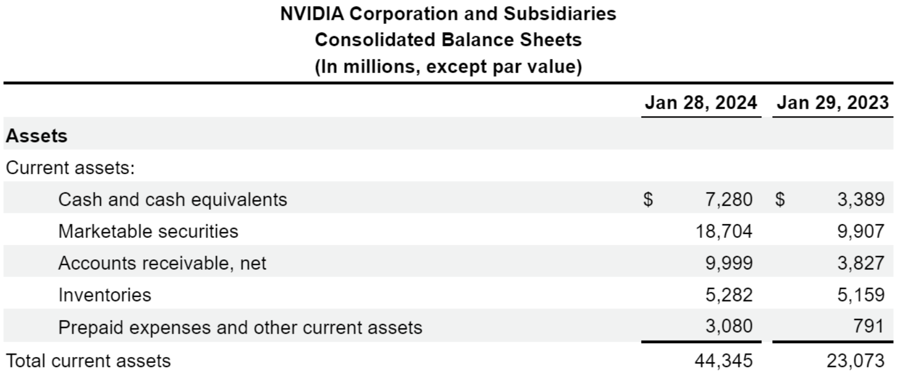
```
▼<context id="c-1">
  ▼<entity>
     <identifier scheme="http://www.sec.gov/CIK">0000320193</identifier>
   </entity>
  ▼<period>
     <startDate>2022-09-25</startDate>
     <endDate>2023-09-30</endDate>
   </period>
 </context>
```

As you can see, this refers to both an entity and a period. Within the entity tag, an identifier is provided. Specifically, the CIK of 0000320193 is provided, which is Apple’s official CIK code. Within the period tag, the start date of 2022-09-25 and the end date of 2023-09-30 is provided. This covers Apple Inc.’s fiscal year.

The *fact* is further associated with a decimals tag of ‘-6’ indicating that the figure is presented in the financials after moving the decimal 6 figures to the left (i.e., in millions). It also includes the id tag of ‘f-69’, which indicates that this is fact #69 within the instance document, and a unit reference tag (unitRef) of ‘USD’ indicating that the revenue is displayed in United States Dollars.

Finally, the figure ‘383285000000’ is displayed, which is Apple’s total revenue for the fiscal year ended on September 30th, 2023 – $383,285,000,000.

Here is another link – Nvidia's 2024 10-K Instance Document – which takes you to the instance document associated with Nvidia’s 2024 10-K filing. All the facts from their 10-K filing, which again was filed as a traditional .hml file, are found within this document. For example, the following image is taken from their balance sheet:



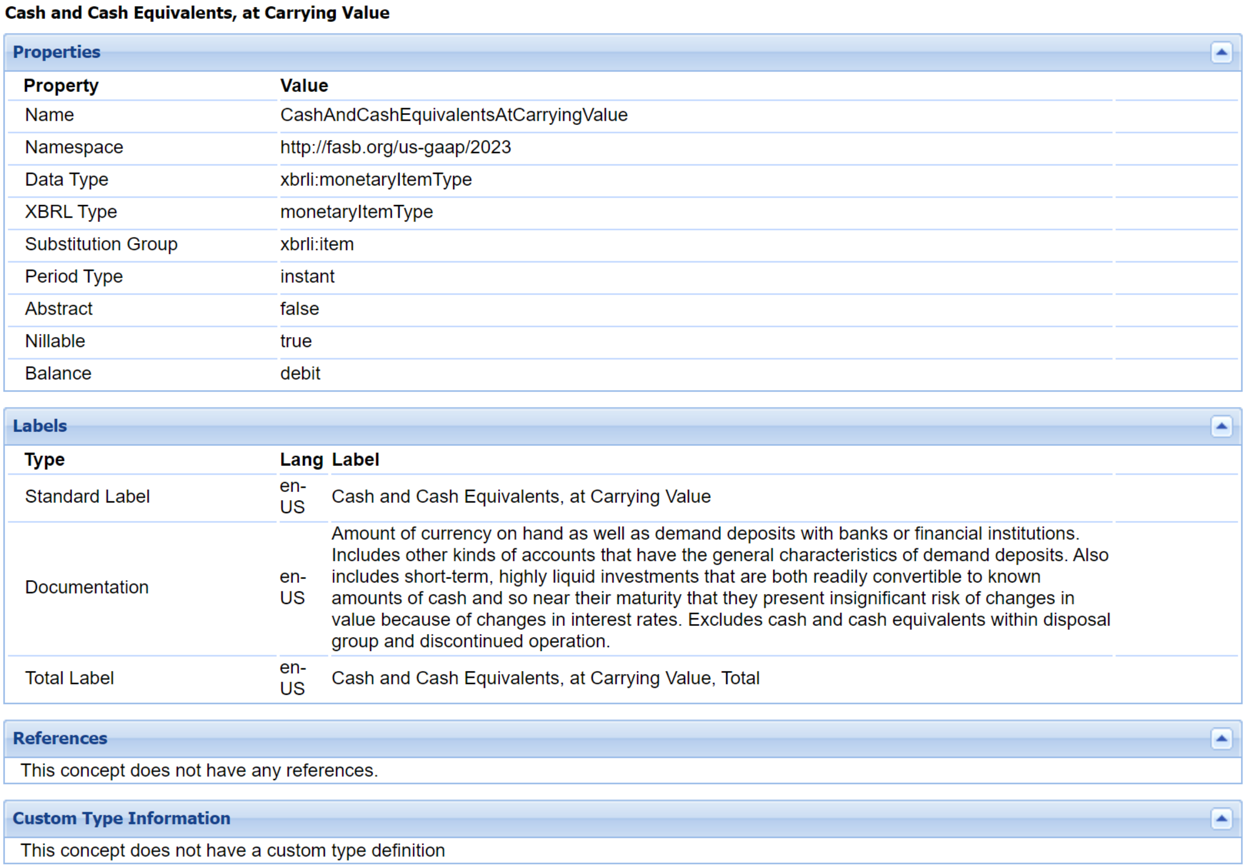
NVIDIA Corporation and Subsidiaries
Consolidated Balance Sheets
(In millions, except par value)

	Jan 28, 2024	Jan 29, 2023
Assets		
Current assets:		
Cash and cash equivalents	$ 7,280	$ 3,389
Marketable securities	18,704	9,907
Accounts receivable, net	9,999	3,827
Inventories	5,282	5,159
Prepaid expenses and other current assets	3,080	791
Total current assets	44,345	23,073

As you can see Nvidia reported Cash and Cash Equivalents of $7,280 million on January 28th, 2024. If you search for this figure within the instance document, you will find the corresponding fact with the following attributes:

```
<us-gaap:CashAndCashEquivalentsAtCarryingValue contextRef="c-9" decimals="-6" id="f-133"
unitRef="usd">7280000000</us-gaap:CashAndCashEquivalentsAtCarryingValue>
```

First, this fact is associated with the reporting tag or concept of ‘us-gaap:CashAndCashEquivalentsAtCarryingValue’. This concept, as defined in the taxonomy, is displayed below.



Cash and Cash Equivalents, at Carrying Value

Properties

Property	Value
Name	CashAndCashEquivalentsAtCarryingValue
Namespace	http://fasb.org/us-gaap/2023
Data Type	xbrli:monetaryItemType
XBRL Type	monetaryItemType
Substitution Group	xbrli:item
Period Type	instant
Abstract	false
Nillable	true
Balance	debit

Labels

Type	Lang	Label
Standard Label	en-US	Cash and Cash Equivalents, at Carrying Value
Documentation	en-US	Amount of currency on hand as well as demand deposits with banks or financial institutions. Includes other kinds of accounts that have the general characteristics of demand deposits. Also includes short-term, highly liquid investments that are both readily convertible to known amounts of cash and so near their maturity that they present insignificant risk of changes in value because of changes in interest rates. Excludes cash and cash equivalents within disposal group and discontinued operation.
Total Label	en-US	Cash and Cash Equivalents, at Carrying Value, Total

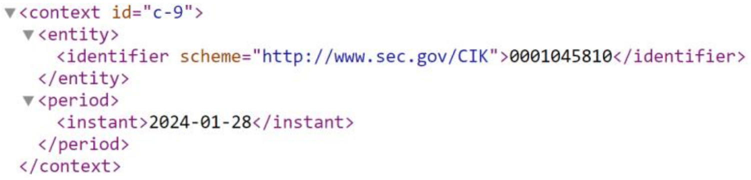
References

This concept does not have any references.

Custom Type Information

This concept does not have a custom type definition

The fact is further associated with the context reference (contextRef) of ‘c-9’. If you search for this within the instance document, you will find the following:



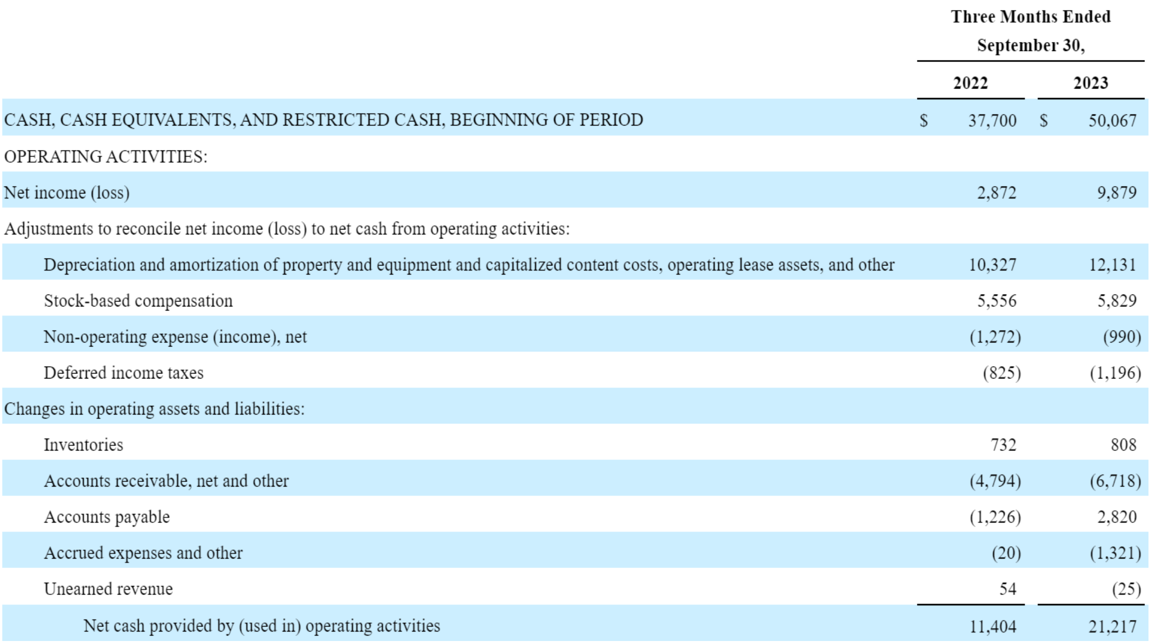
```
▼<context id="c-9">
  ▼<entity>
     <identifier scheme="http://www.sec.gov/CIK">0001045810</identifier>
   </entity>
  ▼<period>
     <instant>2024-01-28</instant>
   </period>
 </context>
```

As you can see, this again refers to both an entity and a period. Within the entity tag, an identifier is provided. Specifically, the CIK of 0001045810 is provided, which is Nvidia’s official CIK code. Within the period tag, unlike the previous example, only an instant is provided – January 28th, 2024. This is because this fact originates from the Balance Sheet which provides data from a specific point in time.

The fact is further associated with a decimals tag of ‘-6’ indicating that the figure is presented in the financials after moving the decimal 6 figures to the left (i.e., in millions). It also includes the id tag of ‘f-133’, which indicates that this is fact #133 within the instance document, and a unit reference tag (unitRef) of ‘USD’ indicating that the cash is displayed in United States Dollars.

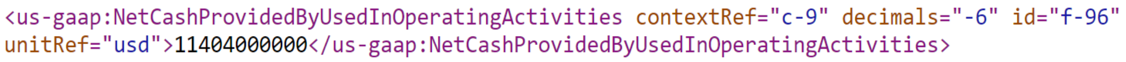
Finally, the figure ‘7280000000’ is displayed, which is Nvidia’s cash on January 28th, 2024 - $7,280,000,000.

Here is a third link – Amazon's 2023 3rd Quarter 10-Q Instance Document – which takes you to the instance document associated with Amazon.com’s 2023 3rd Quarter 10-Q filing. All the facts from their 10-Q filing, which again was filed as a traditional .hml file, are found within this document. For example, the following image is taken from their statement of cash flows:



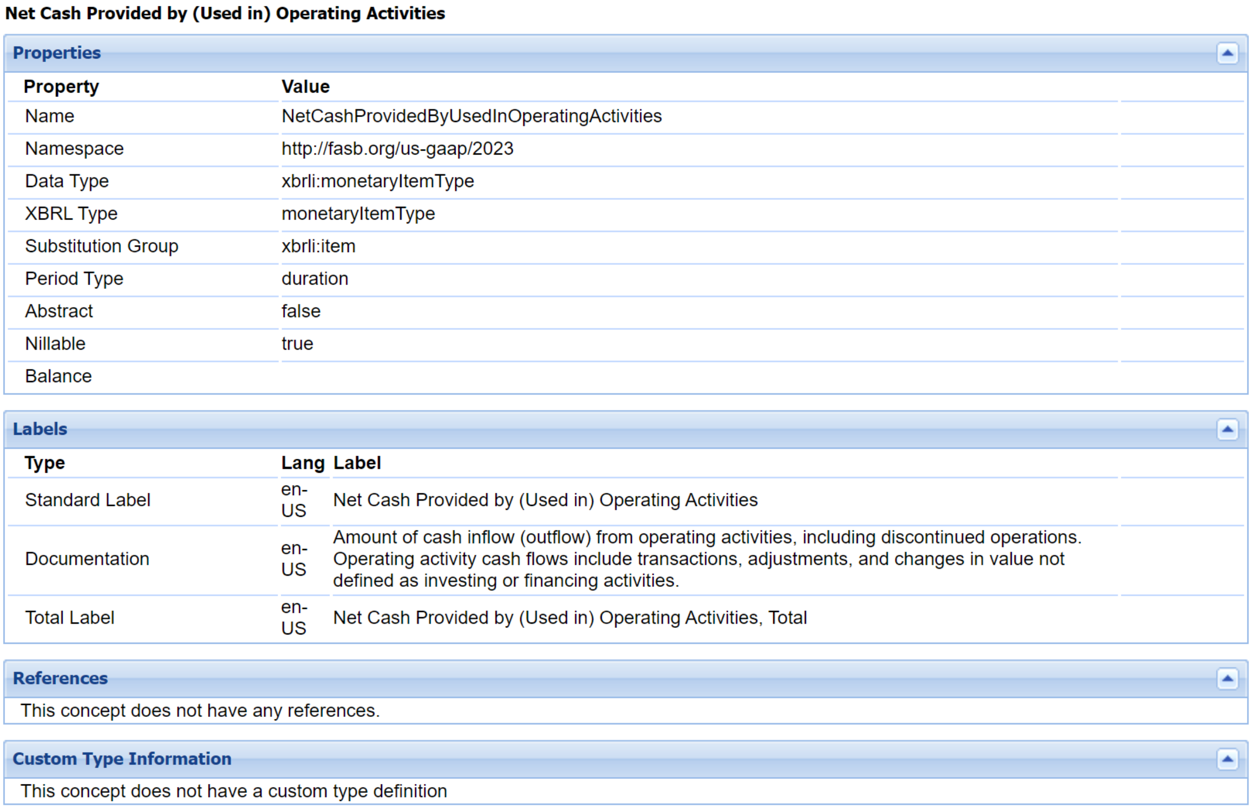
	Three Months Ended September 30,	
	2022	2023
CASH, CASH EQUIVALENTS, AND RESTRICTED CASH, BEGINNING OF PERIOD	$ 37,700	$ 50,067
OPERATING ACTIVITIES:		
Net income (loss)	2,872	9,879
Adjustments to reconcile net income (loss) to net cash from operating activities:		
Depreciation and amortization of property and equipment and capitalized content costs, operating lease assets, and other	10,327	12,131
Stock-based compensation	5,556	5,829
Non-operating expense (income), net	(1,272)	(990)
Deferred income taxes	(825)	(1,196)
Changes in operating assets and liabilities:		
Inventories	732	808
Accounts receivable, net and other	(4,794)	(6,718)
Accounts payable	(1,226)	2,820
Accrued expenses and other	(20)	(1,321)
Unearned revenue	54	(25)
Net cash provided by (used in) operating activities	11,404	21,217

As you can see Amazon reported Net Cash Provided by Operating Activities of $11,404 million for the three months ended on September 30th, 2022 (Note that this is related to the prior year). If you search for this figure within the instance document, you will find the corresponding fact with the following attributes:



```
<us-gaap:NetCashProvidedByUsedInOperatingActivities contextRef="c-9" decimals="-6" id="f-96"
unitRef="usd">11404000000</us-gaap:NetCashProvidedByUsedInOperatingActivities>
```

First, this fact is associated with the reporting tag or concept of ‘us-gaap:NetCashProvidedByUsedInOperating Activities’. This concept, as defined in the taxonomy, is displayed below.



Net Cash Provided by (Used in) Operating Activities

Properties

Property	Value
Name	NetCashProvidedByUsedInOperatingActivities
Namespace	http://fasb.org/us-gaap/2023
Data Type	xbrli:monetaryItemType
XBRL Type	monetaryItemType
Substitution Group	xbrli:item
Period Type	duration
Abstract	false
Nillable	true
Balance	

Labels

Type	Lang	Label
Standard Label	en-US	Net Cash Provided by (Used in) Operating Activities
Documentation	en-US	Amount of cash inflow (outflow) from operating activities, including discontinued operations. Operating activity cash flows include transactions, adjustments, and changes in value not defined as investing or financing activities.
Total Label	en-US	Net Cash Provided by (Used in) Operating Activities, Total

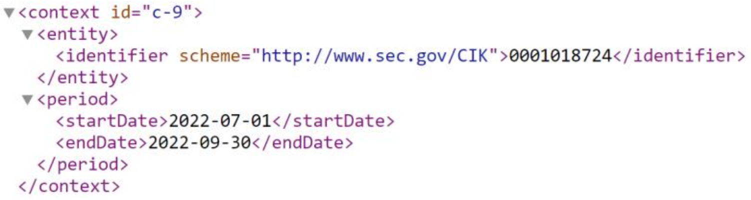
References

This concept does not have any references.

Custom Type Information

This concept does not have a custom type definition

The fact is further associated with the context reference (contextRef) of ‘c-9’. If you search for this within the instance document, you will find the following:



```
▼<context id="c-9">
  ▼<entity>
      <identifier scheme="http://www.sec.gov/CIK">0001018724</identifier>
    </entity>
  ▼<period>
      <startDate>2022-07-01</startDate>
      <endDate>2022-09-30</endDate>
    </period>
  </context>
```

As you can see, this again refers to both an entity and a period. Within the entity tag, an identifier is provided. Specifically, the CIK of 0001018724 is provided, which is Amazon’s official CIK code. Within the period tag, the start date of 2022-07-01 and the end date of 2022-09-30 is provided. This covers Amazon’s 3rd Quarter of fiscal year 2022.

The fact is further associated with a decimals tag of ‘-6’ indicating that the figure is presented in the financials after moving the decimal 6 figures to the left (i.e., in millions). It also includes the id tag of ‘f-96’, which indicates that this is fact #96 within the instance document, and a unit reference tag (unitRef) of ‘USD’ indicating that the cash is displayed in United States Dollars.

Finally, the figure ‘11404000000’ is displayed, which is Amazon’s total increase in cash from operating activities between July 1st, 2022 and September 30th, 2022 - $11,404,000,000.

Who Uses XRBL?

According to XBRL International, more than 600 organizations support the consortium. XBRL is used to serve a variety of purposes including by regulators, individual companies, governments agencies, data providers, analysts, investors, and accountants.

Regulators
Regulators faced with the following challenges can benefit from the use of XBRL:

- Financial regulators that need significant amounts of information about the institutions that they regulate,
- Securities regulators and stock exchanges that analyze the performance and compliance of listed entities and that need this information to be available to investors,
- Business registrars that need to receive and disseminate publicly available data about private and public companies,
- Tax authorities that need information from companies in order to process and review their corporate tax affairs, and
- Statistical and monetary policy authorities that need performance information from many different organizations.

In the United States, some of the most important regulators require their regulated institutions to use XBRL to communicate data. This includes the United States Securities and Exchange Commission (SEC), whose usage of XBRL is discussed in length in the following section; the United States Federal Energy Regulatory Commission (FERC); the Federal Deposit Insurance Corporate (FDIC); among others.

In June of 2019, the FERC began requiring energy companies (i.e., public utilities, electric utilities, natural gas companies, oil pipeline companies, and centralized services companies) to submit their quarterly and annual financial and operational data in XBRL format. Similar to the SEC's adoption of XBRL, these requirements did not change the informational requirements to be submitted to the regulator they simply require that filers tag their data using the FERC taxonomy. The following forms are required to be submitted using XBRL tagging: 6, 6-Q, 2A, 2, 3-Q, 1F, 60, 714, and 1. Data can be downloaded from the following link - FERC Data.

Following the Central Data Repository (CDR) project in 2003, the FDIC in collaboration with the Federal Reserve System and the Office of the Comptroller of the Currency required banks to submit their Call Reports using XBRL in 2006. Call Report data can be downloaded from the following link - Call Report Data.

Companies
Individual entities faced with the following challenges can benefit from the use of XBRL:

- Those that need to provide information to one or more of the regulators,
- Conglomerates that need accurate information within a complex group, and
- Those along a supply chains that need to exchange information to help manage risk and measure activity.

Governments
Governments faced with the following challenges can benefit from the use of XBRL:

- Agencies that are simplifying the process of businesses reporting to government and reducing red tape, by either harmonizing data definitions or consolidating reporting obligations, and
- agencies that are improving government reporting by standardizing the way that consolidated or transactional reports are prepared and used within government agencies and/or published into the public domain.

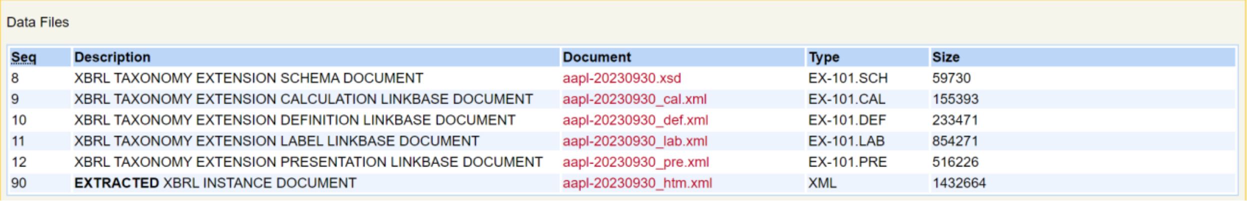
Other Users
Finally the following users can also benefit from the use of XBRL:

- Data Providers that use performance and risk information published into the market and create comparisons, ratings and other value-added information products for other market participants.
- Analysts and Investors need to understand relative risk and performance and need to compare potential investments and understand the underlying performance of existing investments.
- Accountants in support of clients reporting requirements.

XBRL and Financial Reporting with the Securities Exchange Commission in the United States

In April of 2009, the SEC mandated that all public companies subject to filing requirements in the United States provide machine-readable XBRL versions of their quarterly and annual financial reports in addition to the standard text or HyperText Markup language (html) filings as an exhibit to the corresponding filing. These requirements were put into practice in order to make financial information easier for investors to analyze and improve data processing. Collectively, the XBRL files are known as the Interactive Data File and typically include the instance document, schema document, and various linkbase documents.

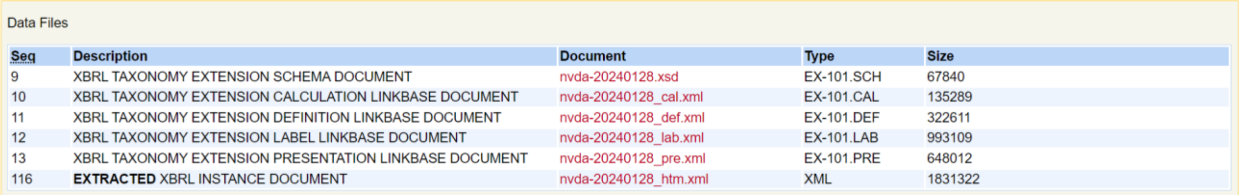
The following image displays the Interactive Data File of Apple Inc.'s 2023 10-K filing (Link).



Data Files

Seq	Description	Document	Type	Size
8	XBRL TAXONOMY EXTENSION SCHEMA DOCUMENT	aapl-20230930.xsd	EX-101.SCH	59730
9	XBRL TAXONOMY EXTENSION CALCULATION LINKBASE DOCUMENT	aapl-20230930_cal.xml	EX-101.CAL	155393
10	XBRL TAXONOMY EXTENSION DEFINITION LINKBASE DOCUMENT	aapl-20230930_def.xml	EX-101.DEF	233471
11	XBRL TAXONOMY EXTENSION LABEL LINKBASE DOCUMENT	aapl-20230930_lab.xml	EX-101.LAB	854271
12	XBRL TAXONOMY EXTENSION PRESENTATION LINKBASE DOCUMENT	aapl-20230930_pre.xml	EX-101.PRE	516226
90	**EXTRACTED** XBRL INSTANCE DOCUMENT	aapl-20230930_htm.xml	XML	1432664

Here is another image displaying the Interactive Data File of Nvidia's 2023 10-K filing (Link).



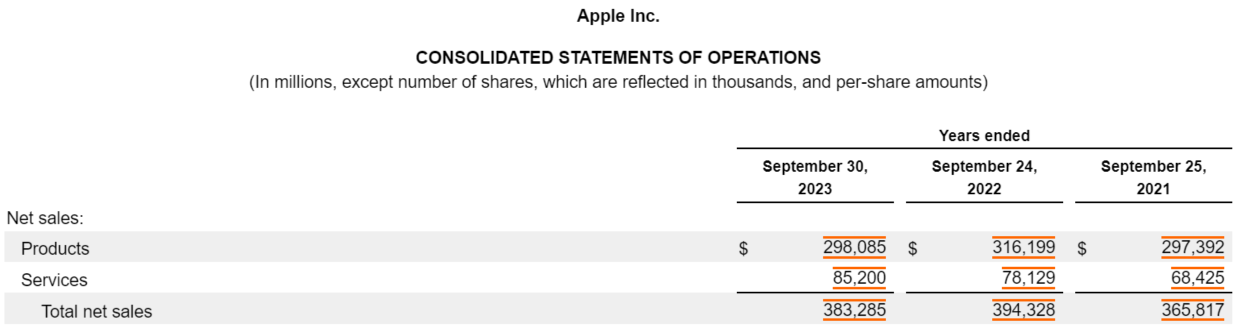
Data Files

Seq	Description	Document	Type	Size
9	XBRL TAXONOMY EXTENSION SCHEMA DOCUMENT	nvda-20240128.xsd	EX-101.SCH	67840
10	XBRL TAXONOMY EXTENSION CALCULATION LINKBASE DOCUMENT	nvda-20240128_cal.xml	EX-101.CAL	135289
11	XBRL TAXONOMY EXTENSION DEFINITION LINKBASE DOCUMENT	nvda-20240128_def.xml	EX-101.DEF	322611
12	XBRL TAXONOMY EXTENSION LABEL LINKBASE DOCUMENT	nvda-20240128_lab.xml	EX-101.LAB	993109
13	XBRL TAXONOMY EXTENSION PRESENTATION LINKBASE DOCUMENT	nvda-20240128_pre.xml	EX-101.PRE	648012
116	**EXTRACTED** XBRL INSTANCE DOCUMENT	nvda-20240128_htm.xml	XML	1831322

When XBRL exhibits are submitted with an EDGAR filing, they are validated for compliance with certain requirements, which produces errors and warning messages when issues are identified. EDGAR also 'renders' the XBRL filings which can then be viewed by humans by clicking the Interactive Data icon on any relevant filing. Here is an example of Apple Inc.'s 2023 10-K filing rendering (XBRL Rendering) and Nvidia's 2023 10-K filing rendering (XBRL Rendering). Specific technical XBRL requirements are located in the Interactive Data File provisions of Regulation S-K; Forms F-10, 20-F, 40-F, 6-K, and N-1A; Rule 405 of Regulation S-T; and the EDGAR Filer Manuel.

XBRL data appears to be widely utilized by various stakeholders. The SEC find that XBRL is used by various stakeholders such as investors, financial analysts, research firms, data aggregators, academics, filers, and SEC staff. During the second quarter of 2017, the SEC identified that XBRL exhibits were accessed more than 53 million times by more than 149 thousand unique IP addresses.

In 2018, the SEC advanced their commitment to XBRL reporting by adopting a proposal requiring the use of the Inline XBRL format. This change was intended to improve the filed data's quality benefiting investors, market participants, and other data users as well as decrease the cost of preparing the data for submission. Inline XBRL allows filers to embed the XBRL data directly into their html filings. The following image provides an example of Inline XBRL for Apple Inc.'s 2023 10-K filing:



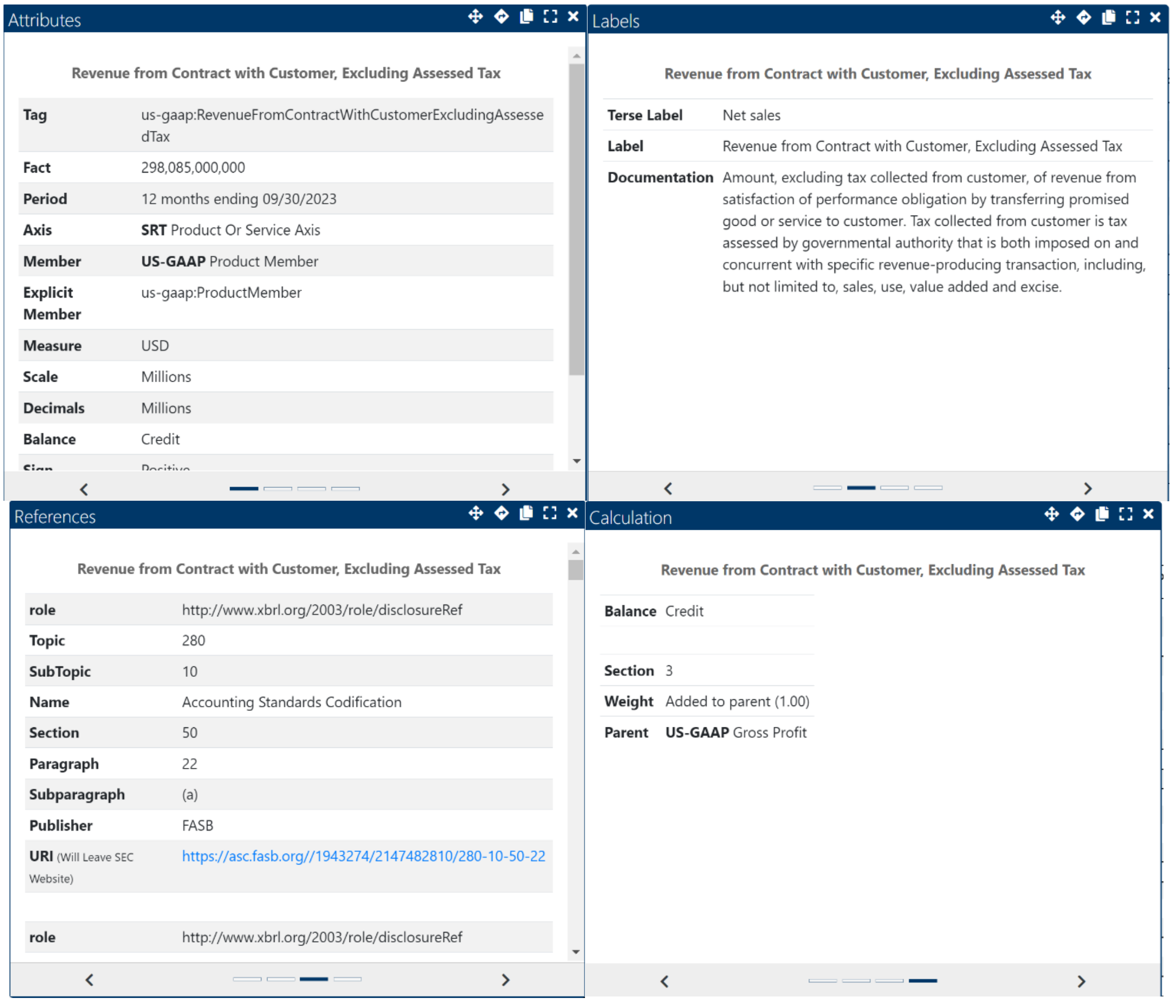
Apple Inc.

CONSOLIDATED STATEMENTS OF OPERATIONS

(In millions, except number of shares, which are reflected in thousands, and per-share amounts)

	Years ended		
	September 30, 2023	September 24, 2022	September 25, 2021
Net sales:			
Products	$ 298,085	$ 316,199	$ 297,392
Services	85,200	78,129	68,425
Total net sales	383,285	394,328	365,817

When you hover over the figures highlighted in orange, you can click on them for more information. For example, when you click on the total net sales for Products for the year ended September 30th, 2023 you see the following:



Attributes

Revenue from Contract with Customer, Excluding Assessed Tax

Tag	us-gaap:RevenueFromContractWithCustomerExcludingAssessedTax
Fact	298,085,000,000
Period	12 months ending 09/30/2023
Axis	**SRT** Product Or Service Axis
Member	**US-GAAP** Product Member
Explicit Member	us-gaap:ProductMember
Measure	USD
Scale	Millions
Decimals	Millions
Balance	Credit
Sign	Positive

Labels

Revenue from Contract with Customer, Excluding Assessed Tax

Terse Label	Net sales
Label	Revenue from Contract with Customer, Excluding Assessed Tax
Documentation	Amount, excluding tax collected from customer, of revenue from satisfaction of performance obligation by transferring promised good or service to customer. Tax collected from customer is tax assessed by governmental authority that is both imposed on and concurrent with specific revenue-producing transaction, including, but not limited to, sales, use, value added and excise.

References

Revenue from Contract with Customer, Excluding Assessed Tax

role	http://www.xbrl.org/2003/role/disclosureRef
Topic	280
SubTopic	10
Name	Accounting Standards Codification
Section	50
Paragraph	22
Subparagraph	(a)
Publisher	FASB
URI (Will Leave SEC Website)	https://asc.fasb.org//1943274/2147482810/280-10-50-22
role	http://www.xbrl.org/2003/role/disclosureRef

Calculation

Revenue from Contract with Customer, Excluding Assessed Tax

Balance	Credit
Section	3
Weight	Added to parent (1.00)
Parent	**US-GAAP** Gross Profit

This provides the user with much more information than just the revenue figure, including the fact attributes, labels used, references, and calculations.

Getting Started with the XRBL's API

In this section, I discuss how to access SEC filing information using XBRL's application programming interface (API). XBRL US provides anyone with access to as-filed SEC or FERC as-filed reports with the XBRL API for no cost; however, XBRL members can enjoy unlimited access without quotas. In the following, Steps 1 through 5 are mandatory with the further steps explaining how to access the data. In future drafts of this guide, a Python program will be utilized to avoid any user interference with parameter inputs.

STEP 1
Create an XBRL US account using your school email address - Registration Link for XBRL US.

STEP 2
In order to access the data, we will use Python. First, we need to select an integrated development environment (IDE) where we will be writing the Python code. There are a number of IDE options (e.g., Programiz, IDLE, Sublime, Atom, Thonny, PyCharm, Spyder, etc.). We will be using Microsoft Visual Studio Code (VS Code). VS Code is a free and open source IDE created by Microsoft. You can download VS Code using this link – VS Download.

STEP 3
Download the Python programming language using Anaconda. Anaconda is a distribution of both Python and R programming languages for scientific computing that aim to simplify package management and deployment. You can download Anaconda using this link - Anaconda Download. Please note that during the download process make sure to check 'add Python to path'.

STEP 4
Obtain Access to XBRL's API. Click the following link to generate an access token – Access Token. Once you are on the site, log into your XRBL account and click the '*Create Client*' button. The site will prompt you to provide a name for the key. You can enter any name (e.g., 'XBRL API'). After clicking submit, you will be provided a Client ID as well as a Client Secret. You must store this information for now.

STEP 5
Within the VS Code environment, create an IPYNB file. Please note that the following section of code was inspired and initially developed by Elizabeth Blankespoor and Ties de Kok. Follow the proceeding code and enter your relevant information…

Import Required Packages

```
import os, re, sys, json, requests, getpass, urllib
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from IPython.display import display, HTML
from datetime import datetime
from urllib.parse import urlencode
```

Obtain API Access

```
password = getpass.getpass(prompt = 'Enter Your XBRL US Password: ')

body_auth = {'username' : ‘ENTER EMAIL HERE’,
             'client_id': ‘CLIENT ID IS FROM XBRL WEBSITE’,
             'client_secret' : ‘CLIENT SECRET IS FROM XBRL WEBSITE',
             'password' : ''.join(password),
             'grant_type' : 'password',
             'platform' : 'ipynb' }
```

```
payload = urlencode(body_auth)
url = 'https://api.xbrl.us/oauth2/token'
headers = {"Content-Type": "application/x-www-form-urlencoded"}

res = requests.request("POST", url, data=payload, headers=headers)
auth_json = res.json()

if 'error' in auth_json:
    print ("Access Denied")
else:
    print ("Access Granted")

access_token = auth_json['access_token']
refresh_token = auth_json['refresh_token']
newaccess = "
newrefresh = "
```

After running the preceding line of code, you must enter your XBRL password and press enter. If you successfully gain access, you will receive the following message, ‘Access Granted’

Access tokens only provide API access for 60 minutes. After 60 minutes you must run the following line of code.

Refresh Access

```
token = token if newrefresh != " else refresh_token

refresh_auth = {'client_id': ‘CLIENT ID IS FROM XBRL WEBSITE’,
                'client_secret' : ‘CLIENT SECRET IS FROM XBRL WEBSITE',
                'grant_type' : 'refresh_token',
                'platform' : 'ipynb',
                'refresh_token' : ".join(token) }

refreshres = requests.post(url, data=refresh_auth)
refresh_json = refreshres.json()
access_token = refresh_json['access_token']
refresh_token = refresh_json['refresh_token']#print('access token: ' + access_token + 'refresh token: ' +
refresh_token)

print('Token Refreshed')
```

API Example

Let Us Gather Non-Restated Revenue and Net Income for Apple, Microsoft, and Amazon for 2022 and 2023
Code for this Example can be found at XBRL US API Example.

Enter Parameters – Elements, Companies, Years, Filings, and Fields

```
XBRL_Elements = ['RevenueFromContractWithCustomerExcludingAssessedTax',
                 'NetIncomeLoss']

Companies = ['0000789019', ## Microsoft (MSFT)
             '0001018724', ## Amazon (AMZN)
             '0000320193', ## Apple Inc. (T)]

Years = ['2022', '2023']

Filings = ['10-K']

Fields = ['entity.cik',
          'entity.name.sort(ASC)',
          'report.filing-date',
          'period.fiscal-year',
          'report.document-type',
          'concept.local-name',
          'fact.value',
          'unit']
```

Run the Query

```
Parameters = {'concept.local-name': ','.join(XBRL_Elements),
              'period.fiscal-period': 'Y',
              'period.fiscal-year': ','.join(Years),
              'unit': 'USD',
              'entity.cik': ','.join(Companies),
              'report.document-type': ','.join(Filings),}

has_dimensions = ‘FALSE’

if has_dimensions == 'ALL':
    dimension_options = ['TRUE', 'FALSE']
else:
    dimension_options = [has_dimensions]

search_endpoint = 'https://api.xbrl.us/api/v1/fact/search'

all_res_list = []
for dimensions_param in dimension_options:
    print('Getting the data for: "fact.has-dimensions" = {}'.format(dimensions_param))

    done_retrieving_all_results = False
    offset = 0

    while not done_retrieving_all_results:
        Parameters['fact.has-dimensions'] = dimensions_param
        Parameters['fields'] = ','.join(Fields) + ',fact.offset({})'.format(offset)
        res = requests.get(search_endpoint, params = Parameters, headers={'Authorization' : 'Bearer
{}'.format(access_token)})

        res_json = res.json()
        res_list = res_json['data']
        all_res_list += res_list

        paging_dict = res_json['paging']
        print('Number of Observations Obtained: ', paging_dict['count'])
        if paging_dict['count'] >= 2000:
            offset += paging_dict['count']
        else:
            done_retrieving_all_results = True

df = pd.DataFrame(all_res_list)
print('Number of Observations: {}'.format(len(res_df)))
```

Keep Relevant Variables and Rename Them

```
df = df[['entity.cik', 'entity.name', 'period.fiscal-year', 'fact.value', 'concept.local-name', ‘report.filing-date’]]
df.columns = ['CIK', 'Company', 'Fiscal Year', 'value', 'account', ‘filing’]
```

Sort Data by CIK, Fiscal Year, Account, and Filing Date and Remove Duplicates

```
df = df.sort_values(by = ['CIK', 'Fiscal Year', 'filing', 'account'])
df = df.drop_duplicates(subset = ['CIK', 'Fiscal Year', 'filing', 'account'], keep = 'last')
```

Scale Values by Millions

```
df['value'] = df['value'] / 1000000
```

Reorganize Data

```
df = df.pivot_table(index=["CIK", "Company", "Fiscal Year"], columns="account", values="value", aggfunc='sum')
df = df.rename_axis(None, axis=1)
df.reset_index(inplace=True)
```

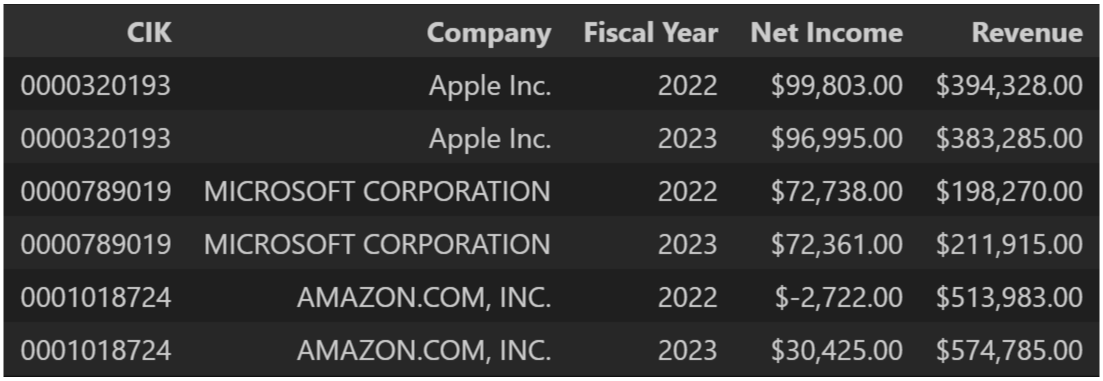
Clean and Format Data

```
df.columns = ['CIK', 'Company', 'Fiscal Year', 'Net Income', 'Revenue']
df['Net Income'] = df['Net Income'].apply(lambda x: f'${x:,.2f}')
df['Revenue'] = df['Revenue'].apply(lambda x: f'${x:,.2f}')
```

Display the Results

```
df
```

They Should Look Like This



CIK	Company	Fiscal Year	Net Income	Revenue
0000320193	Apple Inc.	2022	$99,803.00	$394,328.00
0000320193	Apple Inc.	2023	$96,995.00	$383,285.00
0000789019	MICROSOFT CORPORATION	2022	$72,738.00	$198,270.00
0000789019	MICROSOFT CORPORATION	2023	$72,361.00	$211,915.00
0001018724	AMAZON.COM, INC.	2022	$-2,722.00	$513,983.00
0001018724	AMAZON.COM, INC.	2023	$30,425.00	$574,785.00

XBRL API Exercises

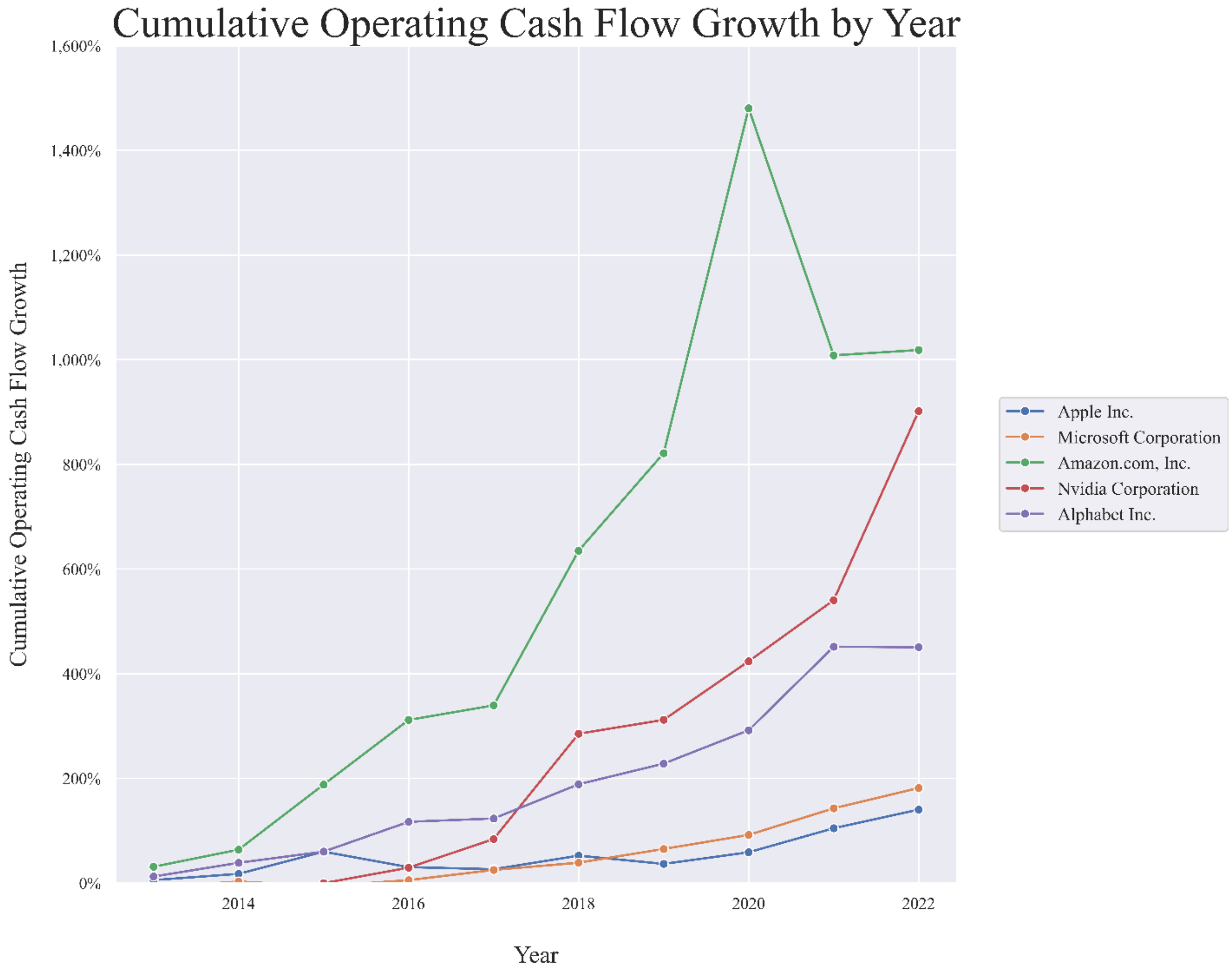
Exercise #1: Cumulative Revenue and Cash Flow Growth – Available at Link.

As of the writing of this exercise, the five largest public companies in the United States by market capitalization are ***Microsoft*** (0000789019), ***Apple*** (0000320193), ***Alphabet*** (0001652044), ***Amazon*** (0001018724), and ***Nvidia*** (0001045810). Let's suppose that we are interested in investing in these companies and want to examine their cumulative revenue and operating cash flow growth over the past decade. Let's use XBRL's API to create graphs for this purpose. I'll start by walking through the procedure for the cumulative revenue growth.

Then, you will repeat the exercise for operating cash flow growth.

Please note that Google was restructured into Alphabet during 2015. To obtain financial data for Alphabet prior to 2015, one must use Google's CIK of 0001288776.

The graph for the operating cash flow growth should look like this…



Exercise #2: Who is Holding the Most Cash? – Available at Link.

Let's suppose that we think Artificial Intelligence (AI) technology is going to radically transform the American economy. We also believe that investing in such technology will require very large capital outlays from firms and that the future lending environment is going to make it prohibitive to successfully fund upcoming projects with new capital raises (i.e., debt or equity issuances). As of the end of 2022, who do we believe will be in the best position to take advance of this opportunity?

To answer this question, we can use XBRL's API to gather the universe of public filers' cash holdings at their fiscal year ending in 2022. We will identify the 10 firms with the most cash on hand. I'll walk through the procedure and then ask you to use a similar procedure to identify which 15 firms lost the most money (i.e., had the lowest net income) in 2022.

Exercise #3: Financial Statement Analysis – Available at Link.

For this exercise, let's perform some financial statement analysis on the car industry, which is associated with the SIC code of 3771. We will rank companies, for their fiscal years ending in 2022, on their (i) current ratio, (ii) debt to equity ratio, and (iii) return on equity. First, I'll walk through how to obtain the current ratios and then ask you to use a similar procedure to identify which firm has the highest return on equity and which has the highest debt to equity.

REFERENCES

RDGFilings. *Federal Energy Regulatory Commission (FERC) XBRL Filing Solution*. Retrieved on April 1, 2024, from https://rdgfilings.com/federal-energy-regulatory-commission-ferc-xbrl-filing-solution/#:~:text=FERC's%20XBRL%20requirement%20requires%20filers,for%20FERC%20forms%20remain%20unchanged.

Ritz, D. 2023. *FERC Commits to XBRL: What's Next for Energy Firms*. Workiva. Retrieved on April 1, 2024, from https://www.workiva.com/blog/ferc-commits-xbrl-whats-next-energy-firms.

United States Securities and Exchange Commission (SEC). XBRL Glossary. Retrieved on December 10, 2023, from https://www.sec.gov/page/osd_xbrlglossary.

United States Securities and Exchange Commission (SEC). 2017. Inline XBRL Filings of Tagged Data. Retrieved on December 30, 2023 from https://www.sec.gov/files/rules/proposed/2017/33-10323.pdf.

United States Securities and Exchange Commission (SEC). 2018. Inline XBRL Filings of Tagged Data. Retrieved on December 29, 2023 from https://www.sec.gov/files/rules/final/2018/33-10514.pdf.

XBRL International. An Introduction to XBRL. Retrieved on March 20, 2024, from https://www.xbrl.org/the-standard/what/an-introduction-to-xbrl/.

XBRL International. FERC Pass XBRL Proposal. Retrieved on April 1, 2024, from https://www.xbrl.org/news/ferc-pass-xbrl-proposal/.

XBRL International. New Ways of using XBRL explored in the US. Retrieved on April 1, 2024, from https://www.xbrl.org/news/new-ways-of-using-xbrl-explored-in-the-us/.

XBRL International. XBRL Essentials. Retrieved on December 12, 2024, from https://specifications.xbrl.org/xbrl-essentials.html.

XBRL International. *What is XBRL?* Retrieved on March 15, 2024, from *https://www.xbrl.org/the-standard/what/an-introduction-to-xbrl/*.

XBRL International. *What is XBRL?* Retrieved on March 12, 2024, from *https://in.xbrl.org/about-us/what-is-xbrl/*.