Consensus and Proof-of-Work – Mining:

A consensus algorithm is a process in computer science by which nodes in a distributed network data store may correctly agree on the value of any particular data item. Traditionally consensus was achieved by the coercive (and corruptible) force of a central authority which simply declared the universal value of each data item. If the central authority was trustworthy and able to secure its decision-making mechanism than all was well. However, even in principle, it is better if the agreement on values is based on an 'objective' and transparent means which functions continuously despite the presence of corrupt or unreliable nodes on the network.

The greatest innovation of Satoshi Nakamota's 2009 Bitcoin blockchain system and paper (https://bitcoin.org/bitcoin.pdf) is a methodology to permit decentralized consensus, therefore obviating the need for central authorities in many disparate fields and domains.

In the context of cryptocurrencies consensus provides the means to validate transactions and prevent the 'double spend problem' (https://www.investopedia.com/terms/d/doublespending.asp) which was seen in one form in the discussion of blockchain and the CAP Theorem in which Jill received Jack's ten dollars but also retained her ten dollars of cryptocurrency which she could then choose to 'spend again' in some new transaction.

There are several categories of consensus algorithms which are being discussed and many variations on each, and there will almost certainly be more. The four main categories are Proof-of-Work (PoW), Proof-of-Stake (PoS), Delegated Proof-of- Stake DPoS) and Byzantine Fault Tolerance (BFT). The mechanism currently used in the real world is PoW.

Proof-of-Work achieves consensus not explicitly by a vote at a fixed moment which decides value and/or validity, but by an emergent process involving participation of a specialized subset of the network participants called 'miners', and entire set of network participants as a whole who verify the 'work' of the miners in order to decide if a block of transactions is valid, and which of a possible group of valid 'mined' blocks is to be added to the chain.

Mining is the process in which pending transactions are collected into a 'block' (structure and characteristics to be specified later) which is used as the input to a 'random' cryptographic hash algorithm (more later) whose output is entered in a sort of 'lottery' whose winner(s) propose the block as the next node in the blockchain, subject to verification of the validity of the lottery ticket number by other users on the network.

Mining a block of transactions entails significant costs of electricity used to run the processors which generate the cryptographic hash lottery ticket numbers. Why would anyone want to spend significant value to run hash algorithms? The inducement for miners to spend funds on electricity to run the hash algorithms is that the eventual lottery winner earns a more than compensating amount of cryptocurrency as well as all the transaction fees paid by participants in the transactions collected in the mined block.

Mining is the mechanism that cryptocurrency system use to create units in a regulated de-flationary manner. For example, Bitcoin halves the amount of coins awarded to a successfully mined block at four year intervals (or approximately every 210,000 added blocks). Initially in 2009 the award was 50 bitcoins per block, which became 25 coins in November 2012, and 12.5 coins in July 2016. By this technique, essentially all bitcoins in existence will be created by 2140, approximately 21 million. In this manner Bitcoin avoids the inflationary tendency of governments controlling fiat currencies such as USD to print money whenever needed to cover massive debt accumulated by reckless politicians who buy votes by distributing printed devalued fiat notes to naive and de-educated voters for 'bread and circuses'.

It should be noted that the decreasing cryptocurrency award is expected to easily be replaced or exceeded from the transaction fees attached to the transactions gathered in blocks. Participants in transactions are motivated to pay more in fees for a transaction in order to make the transaction more attractive to miners to be included in a candidate block which in turn shortens the time required for that transaction to be added to the blockchain and therefore validated. It is also expected that blocksize will increase as mining processor speed increases, thus increasing the number of transactions per block and hence increasing the miner's fee earnings independent of fee amount. These interrelated forces of incentive play a key part in the blockchain mechanism and are described in a mathematical theory of social interaction called 'game theory' (to be discussed later).

Mining and PoW growing of the blockchain consists of four steps: aggregation of a pool of pending transactions into a candidate block, solving the cryptographic problem, validation of the proposed block and solution, assembly of the block at the proposed local parent node and propagation of the validated block to other nodes. The first two steps are carried out by miners and the last two by other general participants on the network and who hold copies of the blockchain.

One further game-theoretic motivation for miners should be mentioned. A block contains a header (to be examined below) which contains the hash of the parent node at which the tentative child block is attached. The choice of the parent is critical. Miners choose the end block of the longest chain in the blockchain, that is, the chain with the greatest accumulation of 'work' and the greatest length. Attaching a candidate block at the longest chain yields the greatest probability that the newly elongated chain will become the validated blockchain (since other miners will choose that candidate block to 'mine' (compete to solve the cryptographic problem associated with the candidate block). The mined (solved) block(s) on the longest chain also attract the greatest number of validatoirs and propagators, thus rewarding the miner first solving the cryptographic problem 'lottery' with the reward of minted coins and collective transaction fees. The inclusion of the hash of the parent block in the header of the child block is how the blocks are 'chained' together, hence 'blockchain', similar in structure to a single-linked-list in software data structures. We will look at the assembly and selection of chains of blocks further on, as well as the temporary difficulty of two or more chains of equal length and proof-of-work called a consistency 'fork' (there are also deliberately induced forks in the blockchain but these are entirely different and will be discussed in future).

The first step in the PoW process is the miner collecting together a set of pending transactions, usually those with greatest attached transaction fees, and creating a block and block header which includes the hash of the block chosen as parent, the tail of the longest chain, the chain containing the greatest accumulation of calculation 'work'. This candidate block contains a special first transaction called the Coinbase Transaction, which rewards the miner with coins into his wallet upon correct solution of the cryptographic problem and inclusion of the candidate block in the blockchain, and a block header consisting of:

[1] timestamp
[2] a version number of the present software/protocol
[3] the hash of the previous 'parent' block
[4] the Merkle Tree root hash (discussed in future)
[5] Target (the form for the goal of the cryptographic problem (usually that the problem solution hash has D leading zeros)
[6] the Nonce (a 4-byte location for a counter used in the problem solution calculation, and initialized to zero)

Mining in Bitcoin consists of calculating the SHA256 hash value of the candidate block and checking to see if that hash has at least D leading zeros, where D is termed the 'difficulty' of the problem. The greater the value of D the greater the difficulty of obtaining a hash with D leading zeros. The SHA256 value is 256-bits long, so the problem mechanism, f(c,x) = SHA256(SHA256(c|x), where c is the 'challenge' (the candidate block header) and x is the nonce value, produces values in {0,1,2,..., 2**256-1}.

A solution to the PoW problem P[f,D] occurs when f(c,x) < 2**224/D.

It can be seen that the difficulty of calculating a solution to P[f,D] increases greatly as D increases, so the 'work' in PoW can be tuned to any cost desired by game-theoretic concerns of motivation and monetary expense. It must be remembered that using a processor to solve a difficult computational problem requires electricity which in turn costs money.

It should also be noted that hash algorithms such as SHA256 produce essentially random results despite the change of a single bit, so certainly for the change of the value of the nonce, and no previous hash result has any bearing on a present result. Solving the cryptographic problem is pure chance, so miners cannot use clever algorithms to gain advantage (well, except perhaps through quantum computing, but quantum computers are very expensive and their algorithms are at an early stage, so for the moment we may overlook the advantage and change of circumstance they may bring).

The basic form of P[f,D] is to run f(c,x) with initial nonce value zero, check whether the result is less than 2**224/D, and if so propose the block as the new blockchain 'tail', and if not, increment the nonce and start again.

The fundamental idea is that finding a solution and attaching a block is very expensive, and since blocks contain the hashes of their parents, the cost of falsifying and necessarily replacing blocks in the previous blockchain is for all practical purposes prohibitive.

It should be noted that the cost to validate a solution for P[f,D] is very very cheap, so it can be carried out by ordinary nodes in the network with essentially no cost. The ease of verification allows the game-theoretic force of the desire for secure and rapid blockchain performance to outweigh the near-zero cost of the validation calculation.

This brings us to the third mining step, validation of the candidate block. Whereas the first two steps were performed only by miners, steps three and four are performed by any node on the network, that is, by ordinary non-mining nodes. Validation of the block consists of checking the following:
[1] the block header hash F(c,x) is less than the target value 2**224/D
[2] the header timestamp is less than two hours in the future
[3] the block size is within limits
[4] the block has correct syntactic structure
[5] each transaction is correct
[6] the first transaction is the Coinbase Transaction

The final step is assembling the block on the local blockchain of the node at the proposed parent block and propagation of the block to other nodes for their validation and assembly. nodes maintain three sets of blocks: the main blockchain itself, branches off the main chain caused by mined blocks with f(c,x) solutions, and 'orphaned' blocks which have f(c,x) solutions but do not have a parent block in one of the first two sets. At any time, each node has a primary 'main' chain which consists of the chain with the greatest accumulated work (and length). When a block is received and validated, the parent hash is retrieved from the header and an attempt is made to attach the block to chain containing the parent (usually at the tail of the main blockchain). If the parent is not found in the main chain or a branch it is considered an 'orphan' and added to the orphans group. If the parent is in the main blockchain than the main chain grows in accumulated work and length. If the parent is in a subchain than a 'fork' has occurred.

A fork occurs when two or more miners solve their cryptogrphic problems P[f,d] at nearly the same time. Let's say miner1 finds a P[f,D] solution using block B1, and miner2 finds a solution using block B2 where B1!=B2. In this case the winners broadcast their solution blocks to their network peer neightbors who validate the solution and propagate the blocks across the network. The fork causes the network to be in a temporary inconsistent state with a subset S1 of nodes with B1 at the main blockchain tail, and B2 at the tail of a subchain, and a subset s2 of nodes with B2 at the main blockchain tail, and B1 on a subchain. Miners who have B1 at the tail of their blockchain will see this chain as the longest and include the hash of B1 in their candidate blocks, and miners with B2 at the tail will include the hash of B2 in their candidate blocks. Usually one group is larger than the other, say S1, which greatly increases the probability of a miner in S1 finding a p[f,d] solution. If block B3 is the candidate block associated with that solution, then B3 will propagate to most or all nodes causing the chain ending in B1-B3 to be the longest and therefore re-establishing consistency.

Even if S1 and S2 are of equal size, it is unlikely that for two blocks running that two miners will find a solution at the same time, so again there will most likely be a solution block B3 which is propagated and resolves the fork. The chance of equal subsets and simultaneous solutions grows much less for K consecutive mining periods for K>2, and Bitcoin has found very few cases of inconsistencies lasting longer two mining periods, and all forks successfully resolved.

Proof-of-Work has proved to be highly successful in regulating by crytographic problems and game-theoretic motivations to maintain, for the most part, a secure immutable blockchain which is rapidly eventually consistent and whose availability delays have not been catastrophically long.

However there are negative to PoW. First is the enormous cumulative cost in energy to run the blockchain. This has prompted interest in the PoS and DPoS mechanisms and there is much present debate. Also, Ethereum has proposed a phased trial introduction of a PoS mechanism operating once every Nth block, where N is proposed to decrease over time.

In addition to the energy problem, it is argued that PoW as presently implemented on the Bitcoin network and elsewhere, is subject to consolidation of miners into central cartels and even monopolies creating opportunities for falsifying blocks by what is loosely termed the '51% attack'. An interesting approach to modernizing the PoW algorithm is proposed by two researchers from University of Luxembourg (and now ZCoin) in their paper 'Egalitarian Computing' (https://arxiv.org/pdf/1606.03588.pdf). Most mining collectives use huge 'asic farms' finely tuned in terms of limited memory and specialized instructions and registers for problems of type P[f,D]. Simply put, the MTP (Merkle Tree Proof) PoW algorithm is designed to require a large memory per cryptographic problem calculation so as to even the playing field between large mining consortia and individuals by making it impossible to use specialized asics for mining, since the cheap asics do not have large memory capacity.